

FINAL REPORT

Forecasting the Relative and Cumulative Effects of Multiple Stressors on At-risk Populations

SERDP Project RC-1541

AUGUST 2011

Joshua L. Lawler
Chad B. Wilsey
Betsy A. Bancroft
University of Washington, School of Forest Resources

This document has been cleared for public release



Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE AUG 2011		2. REPORT TYPE N/A		3. DATES COVERED -	
4. TITLE AND SUBTITLE Forecasting the Relative and Cumulative Effects of Multiple Stressors on At-risk Populations				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S)				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Washington, School of Forest Resources				8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release, distribution unlimited					
13. SUPPLEMENTARY NOTES SERDP Project RC-1541, The original document contains color images.					
14. ABSTRACT The goal of project RC-1541 was to develop and demonstrate a flexible, spatially explicit population model for investigating the effects of multiple stressors on at-risk populations. Demonstration involved using the model to assess the impacts of multiple stressors including climate change, land-use change, training maneuvers, environmental stochasticity, land and pest management, and disease on three different at-risk species on three DoD installations.					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT SAR	18. NUMBER OF PAGES 385	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

Table of Contents

List of Tables	ii
List of Figures	iii
List of Acronyms	ix
Acknowledgments.....	xi
Abstract.....	1
Objectives	3
Background.....	5
Materials and Methods.....	17
Results and Discussion	59
Conclusions and Implications for Future Research/Implementation.....	97
Literature Cited	99
Appendix A. Scientific and Technical Publications	A1
Appendix B. HexSim User's Guide.....	Bi

List of Tables

Table 1. Environmental variables used in constructing Red-cockaded Woodpecker habitat probability models. Data are derived from forest inventory data collected in 2003 at Fort Benning, Georgia, USA.	26
Table 2. Forest management scenarios simulating the effects of loblolly decline on Red-cockaded woodpecker (RCW) habitat. Each scenario assumed a dynamic aging forest.....	37
Table 3. Response and predictor variables and data sources used in constructing a Black-capped Vireo Habitat Model for Fort Hood, TX.....	41
Table 4. Vegetation types on Fort Hood. Those whose growth and change over time are explicitly modeled are in the Fort Hood Vegetation Growth and Disturbance Model are noted with an asterisk.	45
Table 5. Modeled vegetation types for the Fort Hood Vegetation Growth and Disturbance Model and parameter values for the Michaelis-Menton growth equation. Values were estimated for vegetation growing on shallow soils (< 50 cm deep).	46
Table 6. Parameter values for the modified exponential growth equation used to model changes in edge density for the Fort Hood Vegetation Growth and Disturbance Model.	47
Table 7. Scenarios for the Fort Hood Vegetation Growth and Disturbance Model.	50
Table 9. Scenarios for the Vireo-Cowbird Population Model. The scenario including base-wide cowbird control was run with all output scenarios from the Fort Hood Vegetation Growth and Disturbance Model, whereas the no-control scenario was run with a subset.	55
Table 10. Model performance from test data at Fort Benning, GA. Thresholds were selected using the MaxKappa function in the R package PresenceAbsence. Data for ensembles are calculated from weighted averages, followed by values from unweighted averages in parentheses if different from weighted averages. RF = Random Forest, LR = logistic regression, GAM = generalized additive model.	67
Table 11. Model performance from test data at Fort Bragg, NC, USA. Thresholds were selected using the MaxKappa function in the R package PresenceAbsence. Data for ensembles are calculated from weighted average probability, followed by values from unweighted averages in parentheses if different from weighted averages. RF = Random Forest, LR = logistic regression, GAM = generalized additive model.....	67
Table 12. Parameters, their potential ranges, and sensitivity coefficients for the Fort Hood Vegetation Growth and Disturbance Model. Sensitivity coefficients were estimated from 500 model runs using multiple linear regression.	83
Table 13. Probability of a scenario maintaining a given minimum population in the next 50 years (n=15 vegetation model runs).	89
Table 14. Variable importance in the desert tortoise MaxEnt habitat model. Higher values indicate higher importance to the model.....	93

List of Figures

Figure 1. Fort Hood Military Reservation, Bell and Coryell counties, Texas.	10
Figure 2. Historical and projected maximum and minimum annual temperature (a) and annual precipitation (b) for Fort Hood, TX under three CO ₂ emissions scenarios. Data are statistically downscaled climate projections from three global climate models: Community Climate System Model 3.0 (CCSM3), Canadian Global Climate Model 3.1 (CGCM3), and Hadley Center Coupled Model version 3 (HADCM3).	12
Figure 3. Three critical habitat areas (Desert Wildlife Management Areas; DWMAs) within the Western Mojave Recovery Unit. Fort Irwin is also shown for reference.	15
Figure 4. The HexSim modeling framework.	17
Figure 5. Simulating stressor interactions.	19
Figure 6. Red-cockaded woodpecker nest (presence) and random point (absence) locations at Fort Benning, GA. Points represent the geographic center of woodpecker clusters. Circular buffers around points represent the average 109 ha foraging area maintained by RCW groups..	27
Figure 7. Schematic illustrating the various models built for each modeling approach and scale. Each modeling approach is represented by a different color and the different scales are represented by different shapes. The bottom row and far right column illustrate which models were combined for each ensemble.	28
Figure 8. Baseline habitat map for all Red-cockaded woodpecker HexSim simulations. Green indicates high quality habitat, while purple indicates low quality habitat. This map was generated using tree age (≥ 30 and ≥ 60) and tree species (conifer or not conifer).	30
Figure 9. Proposed training ranges reflecting the Transformation at Fort Benning. Although this map does not represent the current plans for range development on post, it was the most current available at the time of scenario development and implementation in HexSim. This map formed the baseline range development scenario from which all other range development scenarios were generated. Values indicate the effect of each range on RCW habitat as the habitat quality was multiplied by the value shown to downgrade the habitat. Map provided by M. Barron.	31
Figure 10. Baseline development habitat map for Red-cockaded woodpeckers (RCW) using proposed training ranges at Fort Benning, GA. Each range was placed on the landscape and habitat within the range was downgraded based on estimated impact of the range on Red-cockaded woodpecker habitat quality. Green areas indicate high quality RCW habitat, while purple areas indicate low quality RCW habitat.	32
Figure 11. Placement of new training ranges under the three Conservation scenarios for Red-cockaded woodpecker (RCW) simulations. Ranges were placed in non-RCW habitat to simulate land use decisions designed to minimize development effects on RCW habitat.	33
Figure 12. Conservation (scenario 1) habitat maps for Red-cockaded woodpecker (RCW) land use change simulations. Ranges were added to the landscape at 10-year intervals over 100 years. Years 2020 (A), 2050 (B), and 2090 (C) are shown. Green colors indicate high quality habitat, while purple colors indicate low quality habitat.	33

Figure 13. Placement of new training ranges under the three Convenience scenarios for Red-cockaded woodpecker (RCW) simulations. Ranges were placed within 400 meters of existing roads.	34
Figure 14. Convenience (scenario 1) habitat maps for Red-cockaded woodpecker (RCW) land use change simulations. Ranges were added to the landscape at 10-year intervals over 100 years. Years 2020 (A), 2050 (B), and 2090 (C) are shown. Green colors indicate high quality habitat, while purple colors indicate low quality habitat.	34
Figure 15. Placement of new training ranges under the three Worst-case scenarios for Red-cockaded woodpecker (RCW) simulations. Ranges were placed in RCW habitat to simulate land use decisions designed to maximize development effects on RCW habitat.	35
Figure 16. Worst-case (scenario 1) habitat maps for Red-cockaded woodpecker (RCW) land use change simulations. Ranges were added to the landscape at 10-year intervals over 100 years. Years 2020 (A), 2050 (B), and 2090 (C) are shown. Green colors indicate high quality habitat, while purple colors indicate low quality habitat.	35
Figure 17. Correlation between April precipitation and Red-cockaded woodpecker (RCW) reproductive success at Fort Benning, GA. Reproductive success was lower in years with more precipitation.	36
Figure 18. RCW habitat maps generated with a “No restoration” forest management scenario for the years 2020, 2050, and 2090.	38
Figure 19. RCW habitat maps generated with a “Restoration” forest management scenario for the years 2020, 2050, and 2090.	38
Figure 20. RCW habitat maps generated with a “Cut” forest management scenario for the years 2020, 2050, and 2090.	39
Figure 21. RCW habitat maps generated with an “Interplant” forest management scenario for the years 2020, 2050, and 2090.	39
Figure 22. Modeling flow-chart for the Fort Hood Black-capped Vireo.	40
Figure 23. Modeled vegetation growth over time following a simulated fire disturbance. Growth rates are defined by a Michaelis-Menton growth equation.	47
Figure 24. Modeled change in edge density following simulated fire disturbances. Boxplots are LiDAR-derived measures of edge density for mapped fires occurring since 1994.	48
Figure 25. Species traits and modeled life cycle for a two-species population model in HexSim. BCVI = Black-capped Vireo, BHCO = Brown-headed Cowbird, HY = hatch year, AHY = after hatch year, ASY = after second year.	51
Figure 26. Spatial data in the Vireo-Cowbird Population Model is represented with hexagonal grids. Portions of the vireo and cowbird breeding habitat maps are shown. Cowbird shooting route added for frame of reference.	53
Figure 27. Map of probability of vireo occurrence produced by the Vireo Habitat Suitability Model (left panel). Grid cells with probabilities below 0.369 are considered unsuitable. Cowbird trapping sites, shooting routes, and breeding habitat modeled as proportional to distance from grasslands (right panel).	56

Figure 28. Empirical relationship between September-March rainfall and survival rates in populations with URTD present in the Western Mojave (data from Peterson 1994, Berry 1997, Christopher et al. 1997, Mullen and Ross 1997).	58
Figure 29. Comparison of predicted presence models from three modeling approaches and the ensemble of the three approaches. Panel A is the predicted presence from a Random Forest model, B is from a logistic regression model, and C is from a generalized linear model. Panel D is the model approach ensemble at the nest scale (weighted average probability of presence). White areas indicate missing or incomplete data.....	66
Figure 30. Comparison of loblolly decline management scenarios for Red-cockaded woodpeckers (RCW) at Fort Benning.....	70
Figure 31. Red-cockaded woodpecker (RCW) population projections under Conservation land-use scenarios.	72
Figure 32. Red-cockaded woodpecker (RCW) population projections under Convenience land-use scenarios.	73
Figure 33. Red-cockaded woodpecker (RCW) population projections under Worst-case land-use scenarios.....	74
Figure 34. Red-cockaded woodpecker population projection under baseline (static precipitation) and projected future precipitation from the CCSM GCM run for the A1B emissions scenario...	75
Figure 35. Threshold independent (AUC) and threshold dependent (overall accuracy and Cohen's κ) measures of model performance for models constructed with only LiDAR-derived measures of vegetation structure at six spatial resolutions (LiDAR), only vegetation and soil variables (VEG & SOIL), and including all predictor variables (FULL). The VEG & SOIL model is built with 25-m resolution presence/absence data and 75-m resolution vegetation and soil data. The FULL model includes 25-m resolution LiDAR and presence/absence data and 75-m resolution vegetation and soil data. Dashed lines mark bounds for relative measures of model fit for AUC (Swets 1988) and κ (Landis and Koch 1977) and an arbitrary minimum acceptable threshold of 75% accuracy.....	77
Figure 36. Mean decrease in accuracy as a measure of variable importance for each predictor variable the 25-m resolution LiDAR model and multi-scale VEG & SOIL and FULL models. Descriptions of variables and their names are provided in Table 3.....	78
Figure 37. Black-capped Vireo presence/absence data and predicted probability of presence based on suitability models constructed with 25-m resolution LiDAR-derived measures of vegetations structure (LiDAR), 25-m resolution presence/absence data and 75-m resolution measures of vegetation functional type and soil depth (VEG & SOIL), and all variables combined (FULL).	79
Figure 38. Current vireo habitat quality (year 0) and output from the Fort Hood Vegetation Growth and Disturbance model for simulated years 15, 30, and 45. Output is from a single run of the Larger Fires scenario.	81
Figure 39. (a) Boxplots of predicted total area of vireo habitat in 2060 from 30 simulation runs for each of six scenarios. (b) Boxplots of predicted area of vireo habitat with a probability of vireo occurrence greater than 0.95. Vertical dashed lines mark the present estimates of suitable vireo habitat from the Vireo Habitat Suitability Model.....	82

Figure 40. Modeled vireo and cowbird abundances in the calibrated Vireo-Cowbird Population Model. The modeled vireo population increases rapidly in response to base-wide cowbird control beginning in 1990, reaching a plateau in 2005 that falls within the confidence limits of the observed vireo population estimates for 2005 and 2006 (A). When modeled cowbird trapping on the west range is stopped in 2005 (B), the modeled vireo population decreases to an equilibrium abundance that falls within the confidence limits of the observed vireo population estimates for 2009 and 2010.	85
Figure 41. Measures of model parameter's direct (μ^*) and interaction effects (σ) on the number of modeled vireos (a) and cowbirds (b) from the Vireo-cowbird Population Model based on the Morris method for sensitivity analysis. Parameter codes: bcPro = vireo productivity, bcJuvSurv = vireo juvenile survival, bcProP = proportion of total vireo productivity assigned to a parasitized nest, bcParSus = percent of vireos susceptible to parasitism by cowbirds, bhOutSurv = cowbird survival out of trap range, bhThresh = range of trap effectiveness, bhRes= minimum resources required for a simulated cowbird to establish a breeding territory, bhHa = cowbird target territory extent, bhPro= cowbird productivity, bhInSurv = cowbird survival in range of traps.....	86
Figure 42. Moving averages of modeled vireo populations from the Vireo-Cowbird Population Model. Each scenario is based on 75 model runs from 15 vegetation projections output from the Fort Hood Vegetation Growth and Disturbance Model. Population model run with (a) and without (b) base-wide cowbird control.	87
Figure 43. Annual projected vireo population abundances summarized with boxplots displaying the median and full range of predicted vireo abundances across all 75 model runs from 15 vegetation projections. Current Training (a), Larger Fires (b), Fire Suppression (c), and Larger Fires without Cowbird Control (d) scenarios are represented. Horizontal line marks the 2010 vireo abundance estimate for Fort Hood.....	88
Figure 44. Habitat model for desert tortoise generated using MaxEnt. Presence data are from deserttortoise.gov and records maintained by Fort Irwin, CA. Green colors indicate high habitat probability, while purple colors indicate low habitat probability.	93
Figure 45. Effects of climate and URTD on desert tortoise density in the Western Mojave.	94

List of Acronyms

AHY	After hatch year
ASY	After second year
AUC	Area under the (receiver operator characteristic) curve
A1B	mid-range greenhouse-gas emissions scenario
A2	mid-high greenhouse-gas emissions scenario
BCVI	Black-capped Vireo
BHCO	Brown-headed Cowbird
BRAC	Base realignment and closure
CCSM3	National Center for Atmospheric Research Community Climate System Model 3.0
CGCM3	Third Generation Coupled Global Climate Model
CF	Cforest (R software package for building random forest predictor models)
CONUS	Continental United States
CRU	Climate Research Unit
CSV	Comma separated variable (file format)
DEM	Digital elevation model
DoD	Department of Defense
DT	Desert tortoise
DWMA	Desert wildlife management area
ELISA	Enzyme-linked immunosorbent assay
EPA	Environmental Protection Agency
FHVM	Fort Hood vegetation growth and disturbance model
GAM	Generalized additive model
GCM	General circulation model
GIS	Geographic information system
GUI	Graphical user interface
HADCM3	Hadley Centre Coupled Model, version 3
HBf	HexSim barrier format (a HexSim model file format)
Hxn	HexSim hexmap file format (a HexSim model file format)
HY	Hatch year
LF	Live fire

LiDAR	Light detecting and ranging
LPJ	Lund-Potsdam-Jena vegetation model
LPJ-GUESS	Vegetation model developed from LPJ by Smith et al. 2001
LR	Logistic regression
MaxEnt	Maximum entropy model
NTC	National training center
PATCH	Program to assist in tracking critical habitat
PBP	Potential breeding pairs
PCMDI	Program for climate model diagnosis and intercomparison
PD	Pine decline
PDF	Portable document format (file format)
PDSI	Palmer drought severity index
PFT	Plant functional type
RCW	Red-cockaded Woodpecker
RF	Random Forest predictor model
ROC	Receiver operator characteristic
SAR	Species at risk
SHP	Shape file (ArcGIS and Arc View file format)
SSURGO	Soil survey geographic
TES	Threatened and endangered species
TNC-FHP	The Nature Conservancy Fort Hood Project
TXT	Text file format
URTD	Upper respiratory tract disease
USDA-	United States Department of the Interior National Resource Conservation Service
NRCS	
USFWS	United States Fish and Wildlife Service
USGS	United States Geological Survey
VCPM	Vireo-cowbird population model
WCRP	World Climate Research Programme

Acknowledgements

The authors wish to thank our collaborators, N. Schumaker, A. Brookes, and S. Shafer. The authors are also indebted to R. Kostecke, D. Cimprich, C. Reemts, M. Noble, S. Summers, and C. Pekins of Fort Hood, TX for sharing data, expert opinion, and constructive feedback, as well as to M. Barron, W. Harrison, T. Marsden, H. Balbach, and P. Swiderek M. Burton for access to and interpretation of the Fort Benning data, and to K. Crawford for access to Fort Bragg data and B. Shomo and C. Darst from USGS for help with desert tortoise case study. Lastly, the authors acknowledge the modeling groups, the Program for Climate Model Diagnosis and Intercomparison (PCMDI), and the WCRP's Working Group on Coupled Modelling (WGCM) for their roles in making available the WCRP CMIP3 multi-model dataset. Support of this dataset is provided by the Office of Science, U.S. Department of Energy.

Abstract

Objectives

The goal of project RC-1541 was to develop and demonstrate a flexible, spatially explicit population model for investigating the effects of multiple stressors on at-risk populations. Demonstration involved using the model to assess the impacts of multiple stressors including climate change, land-use change, training maneuvers, environmental stochasticity, land and pest management, and disease on three different at-risk species on three DoD installations.

Technical Approach

Project RC-1541 had three phases. The first involved model development and specifically developing new code to modify EPA's Program to Assist in Tracking Critical Habitat (PATCH). The new code converted a spatially explicit, individual-based, population model into a flexible population-modeling software platform capable of modeling a wide range of animal species, environments, and stressors. Specifically, the model, now named HexSim, was adapted to 1) simulate complex interactions among stressors, 2) simulate interactions between multiple wildlife populations, and 3) produce a set of easily interpreted outputs and reports. In the second phase of the project, the improved model was parameterized and run for the Red-cockaded Woodpecker at Ft. Benning, the Black-capped Vireo at Ft. Hood, and the desert tortoise at Ft. Irwin to evaluate the relative and cumulative impacts of military activities, environmental stochasticity, anticipated climate change, and other species- and site-specific threats to three at-risk species at three DoD sites. The last phase of the project was technology-transfer in which the model was presented to the DoD Conservation Committee and made publically available via the web.

Results

1. Project RC-1541 has produced a flexible population-modeling platform capable of modeling multiple interacting species and stressors. The model is also capable of simulating dynamic landscapes, inherited traits (e.g., genetics), and a wide array of species.
2. Climate- and vegetation-change projections indicate that temperatures are projected to increase across all three bases. Precipitation and vegetation changes are more variable across climate-change scenarios and bases.
3. Habitat models were developed for all three species for the three bases.
4. Simulated Red-cockaded Woodpecker populations were more susceptible to modeled effects of land-use change than to the modeled effects of climate change. Simulations also demonstrated the importance of recent longleaf pine restoration efforts and the potential for decline of the woodpecker population without continued restoration efforts.
5. Modeling results for the Black-capped Vireo stressed the importance of cowbird control and the potential positive effects of climate-induced increases in fire.
6. Simulations for the desert tortoise population project a continued rapid decline of the Ft. Irwin population in response to upper respiratory tract disease.

Benefits

The main benefit of project RC-1541 is the now-publically available HexSim modeling tool that the DoD and others can use to develop population models for a wide range of animal species in a diversity of environments. The model can be used for exploring alternative development scenarios, implications of management actions, climate impacts, effects of exotic

species and diseases, as well as more generally for risk and population viability assessments. The project also provides guidance for managing three populations of at-risk species.

Objectives

Project RC-1541 directly addressed Statement of Need number SISON-07-01 by developing, applying, and providing methods for assessing the cumulative effects of multiple stressors on threatened and endangered species (TES) and species at-risk (SAR) on military installations. The main objective of the models was to develop a flexible, spatially explicit, population-modeling platform, capable of evaluating both the relative and cumulative impacts of multiple stressors on animal populations. The platform was designed to allow the user to build models capable of assessing the effects of life-history traits, behavior, environmental stochasticity, climate change, spatially and temporally explicit management actions, military activities, land-use changes, and other stressors. A second objective of the project was to use the models to evaluate the impact of multiple stressors on populations of three at-risk species on three DoD bases. The model was used to simulate populations of Red-cockaded Woodpeckers (RCW) at Fort Benning, GA; Black-capped Vireos (BCVI) at Fort Hood, TX; and desert tortoises (DT) at Fort Irwin, CA. A final objective was to present the model to DoD scientists and to make the model publically available. The project had three main technical objectives, with associated specific goals.

Objective 1: Extend an existing spatially explicit population model, enabling it to assess the effects of multiple interacting stressors on population dynamics.

Objective 1 addresses statement of Need (SON) number SISON-07-01 by *developing* methods for assessing the cumulative effects of multiple stressors on threatened and endangered species, and species at risk, on military installations. Objective 1 directly addressed this need by extending the population model PATCH to create the HexSim computer simulation model. The HexSim model is a general, spatially-explicit, multi-species, individual-based life history simulator specifically engineered for evaluating the cumulative effects of multiple stressors on at-risk species.

Objective 2: Evaluate the relative and cumulative impacts of climate change, environmental stochasticity, military activities, and other species- and site-specific threats on three at-risk wildlife populations, at three DoD sites.

Objective 2 addresses SON number SISON-07-01 by *applying* methods to assess the impacts of multiple interacting stressors on at-risk species on DoD installations. Objective 2 governed the project's empirical and applied modeling work, which centered on three separate DoD installations and three at-risk wildlife populations. These case studies examined multiple interacting natural and anthropogenic stressors including climate change, intra-specific competition, and military training activities, plus the impact of other disturbance regimes, on the persistence of at-risk wildlife on multiple DoD properties.

Although the three case studies examined to address Objective 2 were not explicitly designed to test basic hypotheses, they did test several predictions.

(a) Climate change and land-use change have the potential to greatly reduce the Red-cockaded Woodpecker population at Fort Benning.

(b) Brown-headed cowbird control is critical for maintaining the current size of the Black-capped Vireo population at Fort Hood.

(c) Climate change will likely alter the fire regime at Fort Hood in such a way as to increase available habitat for Black-capped Vireos.

(d) The desert tortoise population at Fort Irwin will be negatively impacted by the interacting effects of climate change and upper respiratory tract disease.

Objective 3: Provide the DoD with the improved, highly flexible, user-friendly modeling tool, a comprehensive user's manual, and a training workshop on using the model.

Objective 3 addresses statement of Need (SON) number SISON-07-01 by *providing* methods for assessing the cumulative effects of multiple stressors on threatened and endangered species, and species at risk, on military installations. Objective 3 guided the project's technology transfer activities, which were focused on the delivery to DoD of the HexSim model, documentation, and worked examples that support its use.

Background

Species at risk of extinction often face multiple threats (Lawler et al. 2002). Habitat loss and fragmentation, exotic species, pollution, over-exploitation, and disease all have major impacts on at-risk species. Habitat loss, the most pervasive threat, impacts over 80% of the at-risk species in the United States (Wilcove et al. 1998). Most habitat loss is the direct result of agriculture, construction, or resource extraction that dramatically alters a landscape. The habitat that remains is often fragmented or impacted by other human activities. Exotic or introduced species are the second most common threat, affecting 49% of the at-risk species in the United States (Wilcove et al. 1998, Wilcove and Master 2005). In the last twenty years, a new, far reaching threat has been identified. Average global temperatures are expected to rise between 1.4 and 5.8 °C in the coming century (Houghton et al. 2001). Temperature changes, together with increased carbon dioxide concentrations and altered precipitation patterns, will affect sea level (Meehl et al. 2005), hydrological cycles (Milly et al. 2005), fire regimes, and ecological systems (Walther et al. 2002, Parmesan and Yohe 2003, Root et al. 2003). Changes in the Earth's climate have already led to shifts in species distributions (Parmesan et al. 1999, Thomas and Lennon 1999) and changes in phenology (Beebe 1995, Crick and Sparks 1999). Furthermore, climate change has been clearly implicated in species extinctions (Pounds et al. 1999).

When a species faces multiple threats, the threats can interact in several different ways. Some threats are likely to act additively, some synergistically, and in some cases the effects of one intense threat may make the others relatively unimportant. Some of the better-documented interactions involve synergistic effects in which together two or more human activities produce a greater impact than a purely additive combination of their individual effects would suggest. For example, road building and construction—which tend to impact at-risk species by altering and destroying habitat—may interact with other stressors. These activities can increase exotic species populations by providing access routes and reducing competition from native organisms (Parendes and Jones 2000, Gelbard and Belnap 2003). As a second example, exotic species themselves may act to further alter natural disturbance regimes, thus compounding the effects of human activities such as fire suppression and grazing (Mack and D'Antonio 1998). Some of the least predicted synergistic effects of multiple threats are likely to be those associated with climate change. The redistribution of species in response to changing climates will mean changes in habitat availability, new competitor and predator-prey interactions, and new exotic species (Schneider and Root 2002). As an example, the synergistic effects of climate change and disease have recently been implicated in amphibian declines (Pounds et al. 2006).

Efficiently managing for the persistence of at-risk species requires an understanding of both the relative and cumulative effects of different stressors on wildlife. Due to the limits of field studies and the inherent difficulties in studying at-risk populations, weighing the relative and cumulative impacts of current and potential future threats requires a modeling approach. To best address the DoD's needs, such a model must be able to forecast future wildlife population trends at multiple military installations, and should not be species or stressor-specific. The model should also be individual-based (to better simulate direct impacts), mechanistic (a requisite for forecasting), and parsimonious (for ease of use, and for data-poor environments). Prior to this study, the model that came closest to fitting these criteria was a model known as PATCH, a mechanistic spatially-explicit, individual-based, population model that incorporates GIS representations of real landscapes (Schumaker 1998). The PATCH model had been constructed at the Environmental Protection Agency and used principally in a large alternative futures study.

In spite of the effort that had been put into the development of PATCH, its application to

real-world management and conservation problems was greatly limited. The PATCH model was not flexible enough to simulate a wide range of wildlife life histories or disturbance regimes. PATCH was also limited because it was a single-species, single-sex simulator that could not capture stressor interactions. Finally, PATCH was not user-friendly enough to be useful for a wide audience. In spite of these limitations, the PATCH source code was carefully constructed, quite large, and of great value as a starting point for a more ambitious model development effort, the development of the modeling platform HexSim.

The Red-cockaded Woodpecker at Fort Benning

The Red-cockaded Woodpecker (*Picoides borealis*; RCW) is a non-migratory species endemic to the Southeastern United States. RCWs exhibit a cooperative breeding strategy that makes the species relatively resistant to natural environmental fluctuations (Conner et al. 2001). This cooperative breeding strategy results in a pool of reproductively mature, but reproductively inactive, individuals called “helpers”. These helpers are often male offspring from previous years (Walters et al. 1988). A family group is composed of the breeding pair plus 0 to >3 helpers. Family groups nest in clusters of large, generally older conifer trees (Rudolph and Conner 1991). RCW abundances tend to be lower in areas with encroaching hardwood mid-story vegetation (Hovis and Labisky 1985, Kelly et al. 1993, Conner et al. 1999). In addition to selecting nesting sites, RCWs maintain large foraging territories surrounding nest trees. They prefer to forage in older, larger conifer trees (Walters et al. 2002) which may provide more prey (Hooper 1996, Hanula et al. 2000).

The Red-cockaded Woodpecker was listed as an endangered species in 1970, largely due to widespread habitat loss (U.S. Fish and Wildlife Service 2003). RCWs are dependent on mature, open, pine woodlands maintained by frequent fire. These pine ecosystems have suffered widespread declines due to logging for timber and agriculture (Conner et al. 2001). The majority of remaining RCW populations are found on federal lands and consequently, the Red-cockaded Woodpecker Recovery Plan identifies national forests and military installations as critical for recovery (U.S. Fish and Wildlife Service 2003).

Fort Benning, located near Columbus, Georgia, has approximately 275 active RCW clusters and is designated as a primary core recovery population (U.S. Fish and Wildlife Service 2003). The approximately 74,000-hectare installation is dominated by loblolly pine (*Pinus taeda*), but historically was likely dominated by longleaf pine (*P. palustris*). Forestry practices have varied since the founding of the installation (reviewed in Doresky et al. 2003); however, current forest management includes prescribed burns on an approximate 3-year cycle and the planting of longleaf pine seedlings. These new management practices are targeted towards maintenance of high quality RCW habitat. Since 1996, all active clusters have been monitored for nesting activity on a yearly basis. The population has been steadily increasing since 1996, partly due to artificial cavity creation by installation biologists and The Nature Conservancy (Doresky et al. 2003).

Forest management practices and training locations are driven in part by the listed status of the RCW. Intense effort is extended to ensure the protection of the RCW while maintaining the mission of Fort Benning as a primary training location for the US Army. Military training in close proximity to RCW clusters does not appear to negatively affect nestling survival or success (Doresky et al. 2001); however, removal of nest trees results in displacement of RCW and can be considered “take” under the Endangered Species Act. Maintaining adequate habitat for both foraging and nesting is paramount to successful management of RCWs. Fort Benning’s recovery

goal is 351 potential breeding groups, which requires enough RCW habitat to support 421 active clusters (Final Biological Assessment 2008).

RCWs are a habitat-dependent species that relies on open pine forest for survival and reproduction. As such, the primary stressors for RCWs are those changes that influence habitat quality. Reduction in habitat availability or quality caused by land use changes are of primary importance for the RCW. In addition, recent dramatic declines in loblolly pine in the American Southeast have resulted in concerns over potential habitat loss at Fort Benning from loblolly pine mortality. Finally, climate change has the potential to influence pine tree distribution and recruitment, thereby also impacting habitat availability. Changes in precipitation due to climate change may also negatively impact RCW reproductive rates. Habitat loss and changes in reproductive output have the potential to negatively affect RCW recovery goals at Fort Benning.

Project RC-1541 investigated the effects of three stressors on RCW populations using the HexSim modeling platform. First, the effect of land-use changes on RCWs was modeled through the simulated development of new training ranges under three assumptions of site choice for new ranges. Second, the potential effects of loblolly decline on RCW were simulated, along with three alternative planning strategies to prevent large-scale loss of RCW habitat. Finally, the potential effects of climate change on RCW were modeled through changes in vegetation and changes in precipitation.

Modeling Red-cockaded Woodpecker Habitat

The spatial scale at which input data are sampled can have large effects on model predictions and accuracy (e.g., Thuiller et al. 2003). Many habitat models are built using data sampled at only one spatial scale. However, animals likely select and use habitat at different scales (Wiens 1989), and the scale selected for any particular habitat model should reflect the scale at which key ecological and biological processes occur. For example, the attributes associated with nesting habitat may be very different from those associated with migration corridor habitat. Some habitat models have addressed the issue of scale by creating separate models at different scales, but only a few combine multi-scale data into one predictive model (Graf et al. 2005, Battin and Lawler 2006, McComb et al. 2007). When multi-scale processes are important, it is often preferable to create one predictive model that addresses all relevant scales. However, the best approach for combining multi-scale data is unresolved.

The modeling approach used can also affect model predictions and accuracy. Habitat modeling commonly employs one of two approaches: statistical models (e.g., logistic regression) or machine-learning techniques (e.g. artificial neural networks). Although some approaches perform better than others under specific circumstances, no single approach has emerged as the “best” technique (Elith et al. 2006, Lawler et al. 2006a, Pearson et al. 2006). The predictions from different models using the same data can be vastly different (Lawler et al. 2006a) These differences in predicted habitat reflect the correlative nature of model-building approaches (Araújo and New 2007). Each approach uses different mathematical functions to estimate the relationship between occurrence or abundance and the explanatory variables included in the model, which can lead to differences in predictions. Selecting one approach over another can dramatically influence model structure, model predictions, and any resulting habitat maps.

One strategy for addressing the issue of spatial scale and variation in modeling approaches is to create ensembles of models or model predictions. Predictions resulting from different models may all contain useful representations of the true state of the system and ensemble techniques (also called consensus methods) allow for the incorporation of these different representations into the final outcome. Ensembles may combine multiple permutations

of a single model, models built using different approaches, or models built at different scales (Araújo and New 2007). Ensemble techniques can help reduce model-based uncertainty by identifying areas of agreement among model projections. Consensus methods have been used in broad scale species distribution modeling (Thuiller 2004, Pearson et al. 2006, Araújo and New 2007), but have not been applied to finer scale habitat modeling.

Stressors for the Red-cockaded Woodpecker at Fort Benning

The Red-cockaded woodpecker (RCW) is one species where multiple stressors may influence population persistence. RCW are dependent on high quality habitat for survival and reproduction. Thus, any stressor with the potential to alter habitat is likely to also affect RCW populations within that habitat. In addition, stressors that negatively influence vital rates such as reproduction could slow or reverse the recent population growth seen in this species. The primary aim of these simulations was to project RCW population trends in the presence of known or suspected threats to the population at Fort Benning, GA.

Land use at Fort Benning is driven primarily by changes to the training mission. The recent Base Realignment and Closure (BRAC) and other Transformation projects have resulted in large changes to the training mission at Fort Benning. For example, the relocation of the US Army Armor School and Center from Fort Knox, KY to Fort Benning has added approximately 13,000 new personnel. Additional soldiers on post require development and construction of new training ranges, altering the physical landscape of Fort Benning.

Recent declines in loblolly pine are an emerging forest health issue in the Southeastern US. Symptoms include sparse and chlorotic crowns, low stem wood production, and early mortality (Eckhardt et al. 2010). These declines are thought to be caused by a complex interaction between site characteristics, insect infestation, and infection by fungal species in the genus *Leptographium* (Menard et al. 2010), and have been named simply “pine decline” or PD (Eckhardt et al. 2010). Declines appear to be worse where loblolly pine is considered “off-site” and grown in areas previously dominated by longleaf pine (Eckhardt et al. 2010).

Loblolly declines have been observed at Fort Benning and may negatively affect RCW habitat on the installation (Duke et al. 2008). Much of the forested area of Fort Benning is in loblolly pine, and many of the active RCW nests and much of the current foraging habitat is also composed largely of loblolly pine. In 2007, the installation was focusing on replacing loblolly stands with longleaf pine in an attempt to ensure adequate habitat to reach Fort Benning’s management goal of 351 active clusters (M. Barron, personal communication). Although longleaf pine may also be affected by PD, the extent of those declines is not well-understood.

Projected changes in climate have the potential to alter both the physical landscape of Fort Benning and reproductive rates of RCW. Climate change can cause shifts in the distribution of both plants and animals (Parmesan 2006). If projected changes in climate cause a shift in the vegetation at Fort Benning, e.g. replacing conifers with grasslands, RCW habitat availability would decrease. In addition, previous research suggests that precipitation can influence nestling provisioning rates resulting in measurable reductions in nestling survival (Neal et al. 1993, Conner et al. 2005). Total May precipitation was found to be negatively correlated with the number of RCW fledglings produced in Texas (Conner et al. 2005). Thus, changes in climate could reduce both available habitat and the pool of new recruits for breeder and helper roles in family groups.

The aim of the Red-cockaded Woodpecker case study was to explore the effects of range development (land-use change), climate-driven changes in habitat and reproduction, and the current age structure of the pine forests and early loblolly senescence on the RCW population.

The Black-capped Vireo at Fort Hood

The Black-capped Vireo (*Vireo atricapilla*) is a small migrant bird that breeds in north-central Mexico, central Texas, and isolated locations in Oklahoma and winters along Mexico's Pacific slope (Graber 1961, Wilkins et al. 2006). Vireos in Texas and Oklahoma breed in patchy, primarily deciduous shrublands with relatively low juniper (*Juniperus sp.*) cover (Graber 1961, Grzybowski et al. 1994). In drier west Texas these habitats are maintained by the xeric conditions, whereas habitats in eastern Texas and Oklahoma are created and maintained by fire (Wilkins et al. 2006). Breeding habitats in northern Mexico include xeric lowland scrub and shrublands occurring on steep slopes (Wilkins et al. 2006).

Approximately 75% of the known breeding vireo population in Texas and Oklahoma occurs among three sub-populations: Fort Hood Military Reservation (TX), Kerr Wildlife Management Area (TX), and the Wichita Mountains Wildlife Refuge/Fort Sill Military Reservation (OK) (Wilkins et al. 2006). These sites have hosted most vireo monitoring programs as well as research on habitat selection, dispersal, and demographics. Elsewhere vireos occur in small populations located on private lands with limited access. Therefore, high-resolution range-wide information the abundance and distribution of vireos and their habitats is currently lacking both for the US and Mexico breeding populations. Less is known about vireo wintering grounds and records of wintering vireos are sparse (Wilkins et al. 2006).

The vireo was listed in 1987 as endangered under the Endangered Species Act (Tazik et al. 1993) and assigned both a high priority ranking and high potential for recovery. Observations at the time of listing suggested that vireo populations were highly reduced and acutely threatened by land-management practices, forest succession, urbanization, and brood parasitism by the Brown-headed Cowbird (*Molothrus ater*) (Tazik et al. 1993, Wilkins et al. 2006). All of these threats persist, although some have been reduced through implementation of recovery actions (USFWS 2007), such as prescribed fire, cowbird trapping, and protection of habitats through land acquisition and implementation of Habitat Conservation Plans and Safe Harbor agreements. One potential stressor not yet considered is climate-induced change in disturbance regimes.

In 2007, the US Fish & Wildlife Service (USFWS) recommended that the vireo be down-listed to threatened status because of observed increases in the Fort Hood, Kerr, and Wichita Mts./Fort Sill vireo populations and perceived increases in habitat availability throughout the species range (USFWS 2007). Since listing, the three largest populations have increased from estimates of less than 100 breeding pairs to thousands. These gains largely reflect improved habitat management and cowbird control (Wilkins et al. 2006). However, it is unclear whether similar gains have occurred outside of these managed populations. Perceived increases in vireo abundance elsewhere may also result from increased survey effort and improved detection techniques. A comprehensive assessment of vireo occupancy throughout its breeding range does not exist. Therefore, it is unclear whether the observed recovery is range-wide or limited to the three managed populations or whether the observed recovery is dependent on sustained management of cowbird populations. Furthermore, the potential impacts of future global change have not been assessed.

Fort Hood (Figure 1) is an 87,890-hectare military installation in north-central Texas supporting the largest population of Black-capped Vireos under a single management agency (Cimprich and Kostecke 2006). In 1987, the number of breeding male vireos on Fort Hood was estimated at 87 individuals (Tazik et al. 1993) and the rate of vireo nest parasitism by cowbirds was estimated at 90.8% (Eckrich et al. 1999). Nest parasitism by cowbirds was identified as a major threat to the Fort Hood vireo population, and a cowbird population control program comprised of trapping in cowbird foraging areas and shooting in vireo breeding habitat was

initiated. By 1997, the estimated number of breeding male vireos on Fort Hood was 357 and the rate of cowbird parasitism had fallen to 8.6% (Eckrich et al. 1999). Since then, parasitism estimates have been as low as 2.4% in 2003, and the annual estimate of the breeding male population has been as high as 7184 individuals in 2006 (Goering 1998). Although habitat management and a 4000-ha crown fire in 1996 did create new vireo habitat on Fort Hood, managers suspected that the increase in the Fort Hood vireo population was primarily the result of the cowbird control program (Cimprich and Kosteke, personal communication). In 2005, they initiated an experimental cessation of cowbird control activities on the west half of the installation to document the impact of rising cowbird populations on nest parasitism rates and the vireo population as a whole. In question is whether the Fort Hood vireo population is self-sustaining or whether it is dependent on cowbird population control.

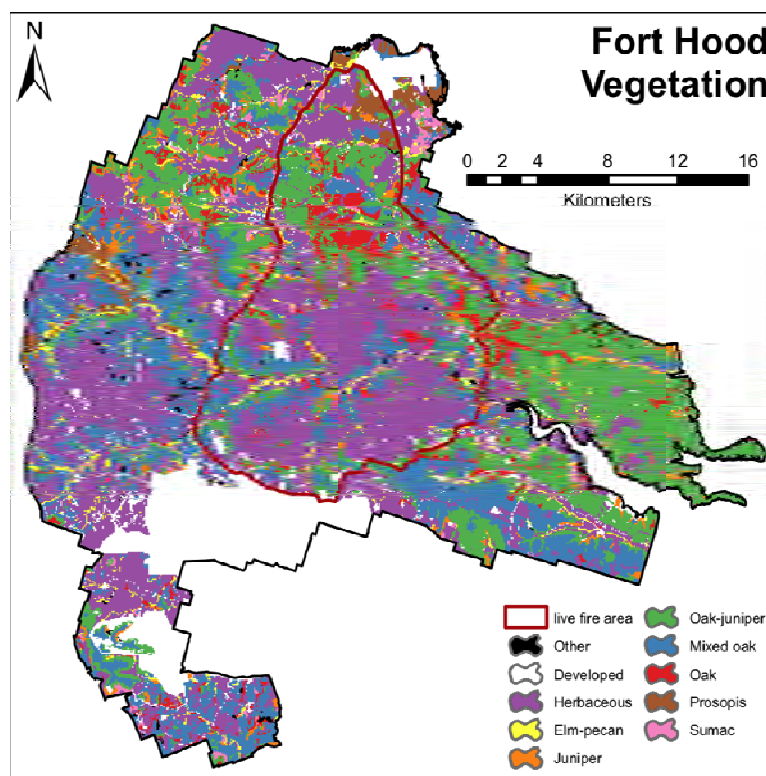


Figure 1. Fort Hood Military Reservation, Bell and Coryell counties, Texas.

Monitoring of the Fort Hood vireo population has been ongoing since 1987 and has included annual point counts, mark and recapture monitoring, and nest monitoring. Because of these efforts, much is known about the demographics of this vireo population. These data provide sufficient information on vital rates and movement to build a spatially explicit individual-based population model (Grimm and Railsback 2005), such as the HexSim model.

Vireo Habitat

In central Texas, the Black-capped Vireo nests in shrub thickets comprised primarily of short deciduous shrub and tree species arranged in clumps on the landscape (Grzybowski et al.

1994). These irregularly shaped thickets are 1-3 m tall, provide 30-60% woody vegetative cover, and are separated by grasslands or rock pavement (Bailey and Thompson 2007). Shrublands may persist on Fort Hood in areas with shallow soils or where fires ignited by artillery are common (Pekins 2006). Otherwise, they represent an early stage in forest succession that given time will transition into taller shrubs and trees (Cimprich and Kostecke 2006). The transient nature of shrubland habitats make them difficult to map and designate for protection or management. Furthermore, as shrubland habitats mature into oak-juniper woodlands, they provide habitat for the endangered Golden-cheeked Warbler (*Dendroica chrysoparia*), also found on Fort Hood (DeBoer and Diamond 2006). Management activities, therefore, must maintain and protect both early- and late- succession habitats on Fort Hood. The RC-1541 project focused on potential changes in the future availability of vireo habitats; however, potential implications for the availability of warbler habitats will also be discussed.

The first and only comprehensive assessment of vireo habitat throughout Fort Hood was conducted in 2002 and 2003 (Cimprich and Kostecke 2006). Areas identified as potential vireo habitat from aerial photos were visited on foot during the 2002 and 2003 breeding seasons to survey for calling vireo males. Areas with calling males or appropriate vegetation were mapped in the field, digitized with a GIS, and merged into a single vireo habitat map for Fort Hood. Since 2003, updates have been made to reflect changes in habitat due to fire or natural succession, and managers are also beginning habitat surveys in the non-breeding season to re-assess habitat patches that have not been revisited since 2003 (Cimprich, pers. comm.). To date, no predictive model of vireo habitat on Fort Hood exists. Such a model would be useful for estimating vireo habitat across Fort Hood for either quantitative analyses or further field-based validation as well as for predicting the potential effect of future land-use change or disturbance on vireo habitats.

New data sources made available since 2003 created an opportunity to produce a predictive model of vireo habitat on Fort Hood. First, vegetation biologists completed a comprehensive vegetation map for Fort Hood in 2007 (Reemts and Teague 2007). Second, Fort Hood contracted LiDAR surveys across the entire installation for natural resource management applications. Thus, for the first time continuous data on vegetation type and height, two factors critical to defining vireo habitat, were available.

LiDAR data were incorporated into a predictive model of potential habitat for the Black-capped Vireo (*Vireo atricapilla*) at Fort Hood, Texas. Objectives were to 1) identify the spatial resolution(s) at which LiDAR-derived measures best predict vireo habitat suitability, 2) assess the relative importance of LiDAR-derived measures of vegetation structure and field-based measures of vegetation composition and soil to model performance, and 3) construct an updated multi-scale vireo habitat suitability model for management applications at Fort Hood using all available data sources. *V. atricapilla* breeds in shrubland habitats historically maintained by fire and shallow soils and persisting 5-30 years (Graber 1961). These transitional habitats are difficult to monitor, making the vireo an appropriate candidate for habitat modeling based on remotely-sensed measures. Furthermore, *V. atricapilla* is an endangered species for which an accurate model of habitat suitability would be useful.

Simulating Changes in Vireo Habitat

Habitat loss due to natural forest succession and land-use change remains a major threat to Black-capped Vireos throughout their range (USFWS 2005). Previous field-based studies (Grzybowski et al. 1994, Bailey and Thompson 2007) revealed that vireos occupy shrublands of

moderate height (1-5m) and with high levels of edge density (>1400 m/ha). These conditions are transitional and in central Texas were historically maintained by fire disturbances. At broad spatial scales, fire disturbance reflects climate (McKenzie et al. 2004a). Climatic change in the past century correlates with an increase in the total area burned in regions of the western US and Canada (Flannigan et al. 2005, Littell et al. 2009). Climate projections for Fort Hood include rising temperatures and little change in precipitation (Figure 2). Data on soils, vegetation types, and climate were used to parameterize the LPJ-GUESS dynamic vegetation model (Sitch et al. 2003) for Fort Hood. Simulations of future vegetation on Fort Hood under multiple CO₂ emission scenarios suggest that fire will be a major driver of vegetation in the future (see Results and Discussion section *Task 2.3. Vegetation-Change Projections*).

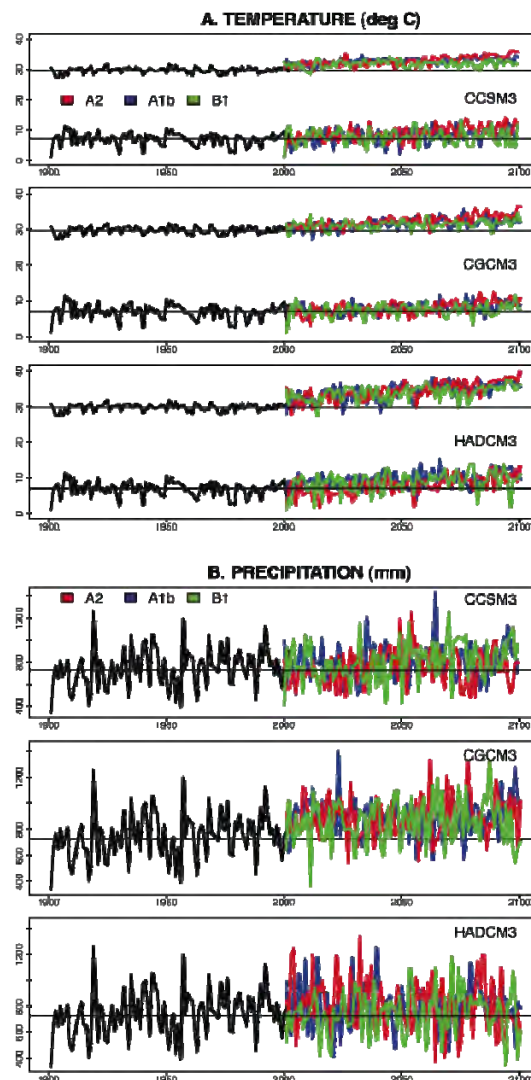


Figure 2. Historical and projected maximum and minimum annual temperature (a) and annual precipitation (b) for Fort Hood, TX under three CO₂ emissions scenarios. Data are statistically downscaled climate projections from three global climate models: Community Climate System

Model 3.0 (CCSM3), Canadian Global Climate Model 3.1 (CGCM3), and Hadley Center Coupled Model version 3 (HADCM3).

In the past, Fort Hood has alternated between fire suppression and “let burn” habitat management policies (Pekins 2006). Ammunition-ignited fires burn annually in the live-fire (LF) region of Fort Hood, but are relatively less common outside of LF. Fuel accumulation during a period of fire suppression combined with drought conditions in 1996 led to a fire that spread out of LF and burned 2108 ha of endangered Golden-cheeked Warbler (*Dendroica chrysoparia*) habitat (Pekins 2006, Noble et al. 2008). The 1996 fire was generally regarded as a failure of the existing management policy because it was too large and dangerous. However, it created vireo habitat throughout its extent and contributed to recent increases in vireo densities. Portions of this seral habitat have already grown too tall to support vireos and the remainder may cease to provide habitat within 15 years. Managers are uncertain whether the “let-burn” policy will maintain sufficient vireo habitat in the future or whether further action is required. Alternative methods of habitat management exist, including bulldozing, mulching, and prescribed fire (Noble et al. 2008), but these are relatively expensive. Projections of the potential impact of fire on Fort Hood in the future would be useful in assessing management needs.

In addition to igniting fires, military training on Fort Hood directly maintains some types of vireo habitat. Shrubland areas that host regular tank training have a unique physiognomy that vireo biologists have named donut habitat (Cimprich and Kostecke 2006) characterized by clumps of tall trees surrounded by shrubs. Regular tank traffic maintains the shrubs around the perimeter of each donut. Vireos occupy the shrubs and generally occur at lower densities than in homogeneous shrublands (Goering 1998). The US Army also thins vegetation periodically to create open spaces for tank maneuvers. Thinning increases the edge density on the landscape and can create vireo habitat where appropriate vegetation exists. The western range of Fort Hood is currently undergoing thinning and there is a potential for future thinning in the eastern range of Fort Hood. Tank training at Fort Hood has recently increased while the installation serves as a staging area for deployments abroad and may change further as a result of the military’s ongoing internal transformation (Pekins, personal communication). The potential impacts of these changes on vireo habitat and the implications for habitat management remain unknown.

Brown-headed Cowbirds

The Brown-headed Cowbird is a brood parasite with multiple hosts (Smith 1999) and an expanding North American distribution (Robinson 1999). Impacts of cowbird parasitism on host populations are variable and most species are able to sustain low to moderate parasitism levels (Smith 1999). However, cowbirds were considered major threats to the endangered Black-capped Vireo, Least Bell’s Vireo (*Vireo bellii pusillus*), Kirtland’s Warbler (*Dendroica kirtlandii*), and the Southwestern Willow Flycatcher (*Empidonax traillii extimus*). Aggressive cowbird trapping and shooting programs facilitated population increases in all four species (Eckrich et al. 1999, Kus 1999, Whitfield and Sogge 1999, DeCapita 2000) and has since been adopted as a tool for managing local songbird populations threatened by cowbirds. Despite this, cowbird trapping and euthanasia remains controversial because cowbirds are a native North American grassland species whose range expansion is a response to anthropogenic land-use changes (Smith 1999). Furthermore, cowbird trapping does not necessarily reduce cowbird population densities and may be required annually at high cost (DeCapita 2000) and although trapping may provide short-term results for host species, it does not address the underlying causes of cowbird range expansion (Hall and Rothstein 1999). Alternative management actions,

such as restoring host nesting habitat (Hall and Rothstein 1999, DeCapita 2000) and decreasing cattle stocking numbers (Goguen and Mathews 2000, 2001, Kostecke et al. 2003) may prove more effective long-term solutions for host species conservation.

On Fort Hood, increases in Black-capped Vireo abundance since 1987 likely resulted from an extensive cowbird control program incorporating both trapping in cowbird foraging habitats and shooting in vireo breeding habitats (Eckrich et al. 1999). Now that estimates of vireo populations on Fort Hood and throughout its breeding range appear to have exceeded recovery goals, the question remains whether vireo recovery is self-sustaining or dependent on cowbird control programs. The USFWS Five-year Review of the vireo cites evidence of declining cowbird population densities within the vireo's Texas range to suggest that cowbird control may no longer be necessary (USFWS 2005). However, cowbird densities appear to be increasing in the Edwards Plateau region around Fort Hood (Kostecke 2008). Therefore, it is unclear if cowbird controls are still required for management of the Fort Hood vireo population. To quantify the impact of cowbird control of vireo populations and nest parasitism rates, managers initiated in 2005 an experimental cessation of cowbird control efforts on the west range of Fort Hood. These data are now available to conduct a quantitative assessment of the effect of cowbird control on vireo productivity and nest parasitism rates.

The Desert Tortoise at Fort Irwin

The desert tortoise (*Gopherus agassizii*) is found in Southern California, Nevada, Utah, Arizona, and Northern Mexico. Desert tortoises are large, long-lived herbivorous reptiles. Individuals take 13-20 years to reach reproductive maturity and adult survival may be critical to population persistence (Doak et al. 1994). Tortoises spend much of the year in burrows, emerging during favorable weather conditions to eat herbaceous vegetation, drink surface water, and breed.

The Mojave population (all tortoises north and west of the Colorado River in Arizona, Utah, Nevada, and California) is genetically, ecologically, and morphologically distinct from the Sonoran and Sinaloan populations found in Arizona and Mexico (U.S. Fish and Wildlife Service 2008). In 1990, the Mojave population was listed as Threatened under the Endangered Species Act, and a recovery plan was published in 1994. The recovery plan identified six recovery units and proposed Desert Wildlife Management Areas (DWMAs) within the recovery units to preserve land believed to be critical to the recovery of the desert tortoise (U.S. Fish and Wildlife Service 2008).

Fort Irwin and the National Training Center (NTC) are located in San Bernardino County, California approximately 65 km north of the city of Barstow. Approximately 9,000 soldiers are stationed or train at Fort Irwin/NTC (<http://www.irwin.army.mil/Post/Info/Pages/FactsandFigures.aspx>). Fort Irwin is located within the Western Mojave Recovery Unit adjacent to the Superior-Cronese DWMA. The Western Mojave Recovery Unit also includes the Fremont-Kramer and Ord-Rodman DWMAs. During the current study, Fort Irwin was issued a No Jeopardy Biological Opinion regarding the southerly expansion of Fort Irwin, which required Fort Irwin to translocate tortoises living in the Southeastern Expansion Area to land purchased off the installation (Walde et al. 2006). Thus, habitat on Fort Irwin is of little importance to desert tortoise recovery, while the area surrounding Fort Irwin is still considered critical habitat. The extent of the current study reflects the importance of off-installation habitat and includes all three DWMAs in the Western Mojave Recovery Unit (Figure 3).



Figure 3. Three critical habitat areas (Desert Wildlife Management Areas; DWMA) within the Western Mojave Recovery Unit. Fort Irwin is also shown for reference.

Rapid declines in desert tortoise populations resulted in the listing of the species as Threatened in 1990. Conservation and recovery of desert tortoises is complicated by the enigmatic drivers of population declines. Although habitat loss has been recognized as a contributor to losses in desert tortoise numbers in recent decades (U.S. Fish and Wildlife Service 2008), other contributing factors to declines have been more controversial. Some authors have suggested that Upper Respiratory Tract Disease (URTD), caused by *Mycoplasma agassizii*, is a primary factor in declines (e.g., Berry 1997). Others suggest that drought cycles may have caused high adult mortality, or that a synergism between drought and URTD may have caused population declines (e.g., Longshore et al. 2003). Predation on neonatal and juvenile tortoises may also play a role in declines (Boarman et al. 2006). However, consensus on the role of these and other factors has not been reached (U.S. Fish and Wildlife Service 2008).

The clinical manifestations and morbidity of URTD varies widely between years and among locations. For example, Schumacher et al. (1997) tested tortoises in Nevada for URTD. Fifty percent of tortoises at this site were seropositive for URTD, while 31% exhibited clinical signs of URTD. However, at another site in Nevada, Lederle and colleagues (1997) found 50% of tortoises seropositive for URTD, but only 2% exhibited clinical signs of URTD. Thus, while similar proportions of these two populations tested positive for URTD, clinical manifestation of the disease varied widely. This variability has complicated the understanding of URTD population-level effects and has led to speculation that environmental factors may be partly responsible for the variation in observed signs of URTD.

One environmental factor likely to play a role in tortoise condition and therefore manifestation of clinical disease signs is water availability. Water availability is a key

determinant of annual plant biomass in the Mojave Desert, the primary food for tortoises (Beatley 1974). Precipitation of at least 25 mm from fall through spring of the following year was required for germination of these winter annuals (Beatley 1974). Over evolutionary history, desert tortoises have been exposed to highly variable water availability. They tolerate widely varying osmotic conditions within their blood plasma (Peterson 1996) and mediate drought conditions by reducing surface activity (Henen et al. 1998). Thus, unless drought conditions have become more severe in recent decades, drought alone is unlikely to have caused the observed population declines. However, anecdotal evidence suggests that drought combined with URTD infection may result in mortality events within populations. For example, Longshore et al. (2003) observed differential mortality at two sites 29 km apart in Nevada in the same year. Both sites had a complete failure of annual winter plant biomass in 1996, but tortoises at the Cottonwood site had much higher mortality. The authors did not test for URTD, but also did not notice clinical signs of the disease at either site. However, it is possible that the differential mortality was due to the interaction between drought and URTD infection at the Cottonwood site.

The aim of the desert tortoise case study in project RC-1541 was to simulate the potential interaction between URTD and drought conditions by projecting desert tortoise population density from the year 2000 to the year 2100 without either stressor, with each stressor alone, and both stressors in combination.

Materials and Methods

Objective 1: Extend an existing spatially explicit population model, enabling it to assess the effects of multiple interacting stressors on population dynamics.

The main activities of project RC-1541 involved developing a flexible spatially explicit individual-based population-modeling platform, HexSim. This goal was broken into three components that were designed to facilitate:

- i. Modeling complex interactions among stressors
- ii. Modeling interspecific interactions
- iii. Developing model-output summaries

The HexSim Modeling Framework

HexSim is an extension of an older model named PATCH, developed by N. Schumaker of the EPA. A significant amount of software infrastructure was re-used in the process of converting PATCH to HexSim. But the most significant features of the HexSim model are entirely new. HexSim is most appropriately characterized as a new and distinct simulation model, the development of which was made efficient through reuse of source code from PATCH.

The PATCH model was spatially-explicit and individual-based. But that tool did not have the flexibility to simulate a wide array of wildlife life histories. It also could not simulate a diverse array of disturbance regimes, nor could it capture interactions between stressors. PATCH was a single species model that had an awkward graphical user interface and very limited reporting capabilities. Finally, every individual was identical in PATCH simulations.

The transition from PATCH to HexSim involved relaxing every one of these constraints. The result was a more powerful and widely applicable tool that can be used by scientists, managers, conservation practitioners, and others.

HexSim is a spatially-explicit, individual-based computer model designed for simulating terrestrial wildlife population dynamics and interactions. HexSim is very general, with landscapes, life histories, disturbance regimes, and most other details being supplied by the user at run-time. HexSim includes a sophisticated graphical user interface (GUI). The model uses spatial data to capture landscape structure, habitat quality, stressor distribution, and other types of information. An illustration of the HexSim modeling framework is shown in Figure 4.

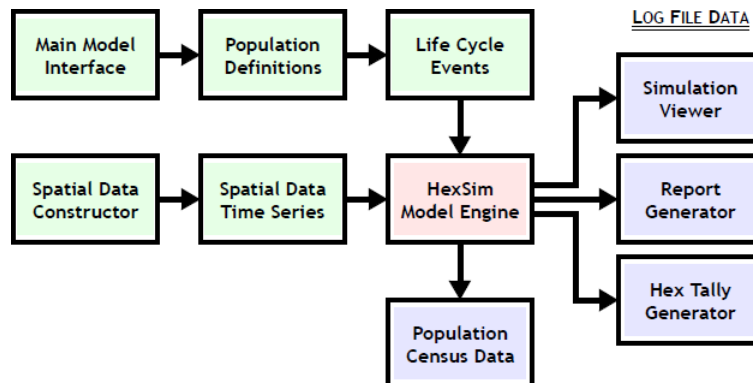


Figure 4. The HexSim modeling framework.

HexSim can work with real or fabricated landscapes. HexSim's design makes it ideal for exploring the cumulative impacts on wildlife populations resulting from multiple interacting stressors. HexSim includes a User's Guide (see Appendix B) which describes each model feature in detail.

HexSim simulations are built around a user-defined life cycle. This life cycle is the principal mechanism driving all other model processing and data needs. Users develop the life cycle when initially setting up a simulation. The life cycle consists of a sequence of life-history events that the user selects from a list. This event list includes survival, reproduction, movement, resource acquisition, species interactions, and many other actions. Through the creative use of events, the user can impose yearly, seasonal, daily, or other temporal cycles on the simulated population. Each event can work with all, or a segment of a population, and events can be linked to static or dynamic spatial data layers. Each life-cycle event has its own data requirements.

HexSim scenarios (its parameterization files) include descriptions of one or more populations, spatial data needs, life cycle definitions, event data, and basic simulation criteria such as the number of replicates and time steps. Each population is composed of individuals, and individuals have traits that can change probabilistically, or based on age, resource availability, disturbance, competition, etc. HexSim also includes optional genetics and heritable traits. The use of traits allows individuals to have unique properties that change in time and space. Traits also allow populations to be segregated into classes, such as males and females, fitness categories, disease categories, etc. Combinations of trait values can be used to stratify events such as survival, reproduction, or movement.

In HexSim, all individuals are placed into one of two categories—floaters or group members. This scheme is useful for simulating territorial species, but it can be effectively ignored when working with non-territorial animals. Group members are territory-holders, and only group members are allowed to reproduce.

The HexSim life cycle consists of events that are drawn from an event list. The event list has a wide array of event types, each of which can be parameterized in many different ways. Simple scenarios may use few events with minimal parameterization and little spatial data. But when more complexity is warranted, HexSim allows a great deal of data and behavior to be added to its simulations.

Traits are a fundamental part of HexSim scenarios. HexSim traits are defined at the population level, but they are implemented on an individual basis. HexSim's traits can be used to control most life cycle events. Traits influence events because events can be stratified by trait combinations. For example, a movement event might be set up to operate only on a fledgling stage class, or a survival event might assign mortalities based on the values of a trait that reflects resource acquisition. In addition, one trait's values can be influenced by multiple other traits, which makes it possible to set up stressor interactions and complex feedback loops. Traits can also be used to capture species' interactions such as parasitism, competition, mutualism, breeding, and so on.

HexSim traits can be probabilistic or accumulated (changing with time or exposure, etc.). HexSim also has a genetics module and heritable traits. Accumulated traits change based on individual experience, which is captured in a series of variables called accumulators. Events called updater functions modify accumulators, which in-turn alter accumulated traits. Transition events are used to modify probabilistic traits. Interaction events change trait values based on intra- and inter-specific interactions. Trait builders automate the process of setting up the most common trait types. Heritable traits are based on an individual's genotype, which is created at

birth from its parents genotypes. HexSim genetics also include mutation, which can further impact an individual's heritable traits.

A number of different data analysis tools are built into the HexSim model. These include reports, output maps that illustrate model dynamics, and an animated simulation viewer. In addition, HexSim Census events can be used to track population size, stratified by any combination of traits. The HexSim reports are written as comma separated variable (CSV) files, and can be opened with a spreadsheet. The output maps are easily converted to raster images or shapefiles. Other usability-related features include a batch processing tool, a sensitivity analysis module, the ability to change color schemes, and more.

The HexSim model is available now at: www.epa.gov/hexsim.

Modeling Stressor Interactions

HexSim's approach to simulating stressor interactions is based on traits. HexSim has probabilistic, accumulated, and genetic traits (see Appendix B -- the HexSim User's Guide). Traits are specified at the population level, but implemented on an individual basis. Each trait must have at least two trait values. Probabilistic trait values are selected based on user-defined probabilities. Accumulated trait values change when the accumulators they are linked to are modified. Accumulators are simply variables that each individual carries with them throughout their life. Genetic traits are each tied to a specific locus, and the trait values are determined by the alleles present at that locus.

One approach to simulating stressor interactions is to use probabilistic traits to link multiple other traits. Figure 5 illustrates this concept. The life cycle depicted in the figure includes a fitness trait that is linked to both disturbance and resource acquisition traits. Resource acquisition is in-turn a function of landscape structure and competition. Survival rates vary based on both the fitness trait values and whether an individual is diseased. But fitness also directly impacts individual's ability to fight off the disease once they become infected. Interactions such as these are easily built into HexSim simulations.

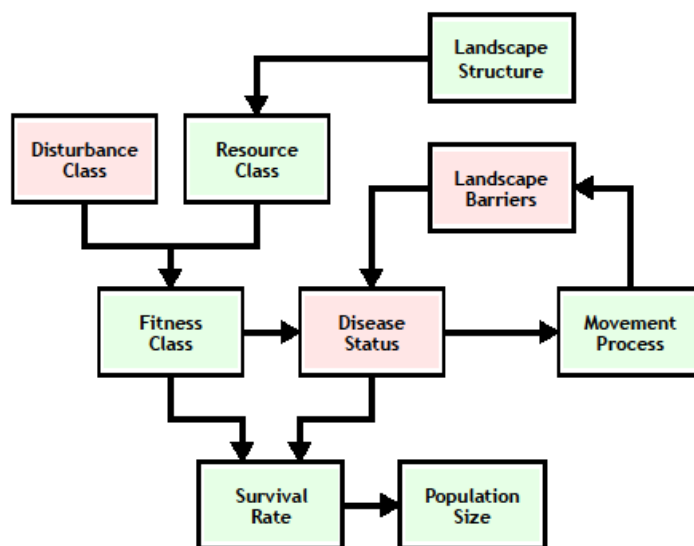


Figure 5. Simulating stressor interactions.

Another way to simulate stressor interactions involves the use of multiple spatial data sets. Many HexSim events involve animal-landscape interactions, and any number of different landscape maps can be built into a single HexSim simulation. Different spatially distributed stressors, represented by unique spatial data, can simultaneously influence a population. For example, two separate dynamic spatial data sets might capture the distribution of a toxic chemical and the distribution of a parasite. As individuals move across the landscape, they may come in contact with one or both. The consequences for survival or reproduction of being simultaneously exposed to the toxicant and parasites could be made greater than the sum of the two, were they independent. Or, exposure to the toxicant might make an individual more susceptible to infection by the parasite.

Modeling Species Interactions

The first step in adding species' interactions was to make HexSim a multi-population model. Once multiple populations could be simulated simultaneously, mechanisms had to be created to allow them to interact. In HexSim, populations can now interact in three different ways. First, HexSim includes an Interaction event. This event is flexible, and users can control the probability that individuals will meet, the probability that they will interact if they do meet, and the types of outcomes that will result from the interaction. Outcomes can include death, group joining, pairing (for breeding purposes, which is usually performed between members of a single population), and number of other consequences. Interaction events are useful for simulating predation, for spreading disease, for forming pair-bonds, and generally for exposing members of one population to the members of another. Since an outcome of interaction can include the modification of trait values, a wide array of consequences may result from individual interactions when mediated by an Interaction event.

The second mechanism for simulating species' interactions involves use of the Generated HexMap event. This event creates new spatial data while a simulation is running. These Generated HexMaps, as they are called, can be built up using map algebra from a combination of existing spatial data sets, and from species' presence / absence information. The latter is gathered in real-time from the simulation itself. The Generated HexMap event also has smoothing and rescaling functions that can be used to create maps with smooth gradients. For example, in a predator-prey simulation a Generated HexMap event might be used to guide predators towards prey. If regions exist in the landscape that predators would not enter, this information could be coupled with the prey occupancy data to enforce realistic dispersal behavior. Predator dispersal would use the Generated HexMap, and be parameterized so as to impart an attraction to areas of high prey density. Populations can also use Generated HexMaps to detect whether individuals are in the presence of members of another population. The results of such queries can be used to modify trait values, and thus can influence vital rates or behavior.

The third mechanism for simulating species' interactions provides a means through which populations may compete for resources. In HexSim, individuals from all populations will have resource targets. Resource targets are stratified by at least one trait, and they specify the maximum resource needs for each individual with the indicated trait values. For example, a population may have a stage class trait with juvenile, subadult, and adult trait values. The resource targets for this population could be stratified by stage class, with juveniles, subadults, and adults being assigned targets of 5, 10, and 15 resource units respectively. Individuals may acquire resources from their group's ranges (equivalent to a defended territory for a territorial species) or from their explored areas (equivalent to a home range for a territorial species). Within a population, ranges may not overlap, but explored areas can. Resource Acquisition updater

functions, used within Accumulate events, partition the available resources up among individuals. The partitioning is influenced by landscape structure, individual resource targets, intra-specific competition (either within ranges, or across explored areas), individual resource priorities (e.g. within a population, healthy individuals may out-compete sick individuals for resources), and inter-specific competition. Inter-specific competition for resources is controlled by a population-specific Competitive Ability parameter. In multi-species simulations, some populations may compete while others do not. And competition may occur between any number of populations simultaneously. The Competitive Ability parameter specifies the access to resources that each population will have relative to every other. Inter-specific competition is mediated on a individual-by-individual basis, so a poor competitor may out-compete another population if it is present in sufficient numbers.

Enhanced Model Outputs

Four classes of simulation products have been developed to both make HexSim more user friendly, and to help users better comprehend the simulation results (see Appendix B -- the HexSim User's Guide). These products include HexSim censuses, reports, map-based tallies, and the tool called the Simulation Viewer. Each is discussed below. When HexSim runs, any Census events present in the Event Sequence will write files directly to the simulation results directory. All other simulation results are saved to an ASCII log file. Each of HexSim's reports and tallies, and all of the data necessary to drive the Simulation Viewer, are extracted from the log file. Thus users do not need to anticipate what simulation products they will require in advance of running HexSim. All such products are generated after a simulation has been completed.

HexSim's census files are the easiest outputs to generate and the simplest to work with. Every Census event will generate CSV (comma-separated variable) format. Users may place as many Census events into the life cycle as they want. Census event output is useful for making plots of population size with time. However, population size is tracked in several different ways in these output files. There are columns for the total population size, the numbers of group members and floaters, and for the number of individual with each combination of the traits selected in the event's interface. Finally, an instantaneous measure of Lambda (the rate of population growth) is provided. The instantaneous lambda value for time step T is equal to $[\text{Population at T}] / [\text{Population at T-1}]$. This simple computation can be cumbersome to perform by hand, so its is included in the CSV file as a convenience.

HexSim reports are accessed through the Scenario drop-down menu. Reports provide a variety of information about what actually happened in HexSim a simulation. There are 12 different HexSim Reports, and every one writes a CSV file intended to be read into a spreadsheet. The Dispersal Steps, Dispersal Paths, and Projection Matrix reports produce data tallies, so users must supply initial and final time steps. The tallies are then gathered only for the specified period. The remaining 9 reports are stratified by replicate and time step, so there is no need to specify initial and final time steps. If there are multiple populations in a simulation, then a CSV report file will be generated for each one. Reports are available to track births and deaths (3 separate reports), genetics, movement (3 separate reports), use of space, barrier impacts, environmental stochasticity, and population interactions. A report is also available that builds a population projection matrix from the simulation output.

HexSim tallies are also accessed through the Scenario drop-down menu. HexSim tallies are collections of count data recorded for each hexagon in a grid. These counts are incremented over all replicates, and all time steps specified by the user. HexSim tally data can be saved in CSV format, or as HexSim HXN (HexMap) files. Both formats can be read by HexSim's Display

Spatial Data tool, and therein displayed as a image, and saved to a bitmap, Shapefile, or CSV file. Tally data can be placed into two functional groups:

Individual-Based Tallies

- Births
- Deaths
- Births minus Deaths
- Interactions
- Occupancy
- Trait Counts

Hexagon-Based Tallies

- Barrier Deaths
- Barrier Deflections
- Barrier Transmissions
- Dispersal Flux

Hexagon-based tallies capture individual interactions with the landscape. When an individual encounters a barrier, it necessarily does so at a specific hexagon. The same is true for Dispersal Flux, which records the number of dispersers that pass through each hexagon. When these interactions take place, the target hexagon's tally gets incremented. Hexagon-based tallies are always attributed to a single hexagon, and they are always integer-valued.

Individual-based tallies record information specific to an individual. This information must then be attributed to one or more hexagons. There are two ways that individual-based tally data can be attributed to hexagons. Every individual can be assigned a single hexagon location. Tally data can then be written to these individual locations. Tally data may also be assigned to group member ranges as a whole. When this is done, the each hexagon in a range of size N will be incremented by $1/N$. Individual-based tallies can thus be integer-valued, or real-valued.

The HexSim tally dialog allows users to select a starting and ending time step. Data will only be tallied for the time steps included within these bounds. This allows users to avoid recording information prior to the simulation's reaching steady state, or to tally data for just a small temporal window. If multiple replicates were run, the tally is performed cumulatively across all replicates, but still respecting the starting and ending time steps.

The HexSim Simulation Viewer is accessed through the HexSim menu. This tool allows users to visualize how groups or individuals moved around in the landscape as the simulation progressed. The Simulation Viewer reads a simulation log file, and generates an animated display much like a movie. Users may choose to show the locations of groups, individuals stratified by trait combinations, or individual movements including dispersal and exploration. The simulation viewer is easy to use, and it provides unique insights into the simulation dynamics.

Additional information, including worked examples of HexSim applications, may be found at the web site: www.hexsim.net.

Objective 2: Evaluate the relative and cumulative impacts of climate change, environmental stochasticity, military activities, and other species- and site-specific threats on three at-risk wildlife populations, at three DoD sites.

Site Visits and Data Compilation

All three sites were visited in Fall 2007. Data from each installation were assembled and wildlife biologists, vegetation biologists, and land managers were consulted about current and future trends on post.

Spatial Data Development

Spatial grids were developed for each of the three study areas. The grids for the Fort Benning and Fort Irwin study areas had a spatial resolution of 1000-m and the grid for the Fort Hood study area had a spatial resolution of 960-m to match the spatial resolution of other datasets that were used to simulate population dynamics of Black-capped Vireos at Fort Hood. The study areas for Fort Benning and Fort Hood extended roughly 5 to 20 km from the installation boundaries. The study area for Fort Irwin extended roughly 20 km to the north and east of the installation and 60 km to the west and south of the installation. The larger study area at Fort Irwin was designated to allow modeling of the desert tortoise population in the habitats surrounding the base.

Climate-Change Projections

Climate-change projections were developed by Sarah Shafer of the USGS (Shafer et al. In-Press). Historical climate data for the period of 1961 to 1990 were downscaled from the University of East Anglia's Climatic Research Unit (CRU) CL 1.0 and CL 2.0 global datasets (New et al. 1999, New et al. 2002) to the spatial grids for each of the study areas using a locally weighted lapse-rate-adjusted interpolation method. The downscaled data included measures of monthly mean temperature (°C), total precipitation (mm), and mean sunshine (%) from the CRU CL 2.0 data set and monthly mean cloud cover (%) from the CRU CL 1.0 data set. All variables were averaged over the 30-year period. A monthly time series was generated for the 20th-century using monthly anomalies for temperature, precipitation, and percent sunshine (based on cloud cover data) calculated for each month in the 1901-2002 CRU TS 2.1 global 30-minute gridded data set (Mitchell and Jones 2005) based on a 1961-1990 30-year mean base period. Temperature anomalies were calculated as differences (each monthly value minus the 1961-1990 30-year mean base period value). Precipitation and sunshine anomalies were calculated as ratios (monthly values were divided by the 1961-1990 30-year mean base period values). These anomalies were interpolated to each study area grid using a geographic-distance-weighted bilinear interpolation method. They were then applied to the CRU CL 2.0 1961-1990 30-year mean value for each climate variable in each grid cell.

Future climate projections for each study area were generated by downscaling projected climate simulations from three general circulation models (GCMs), including the CCSM3 (Collins et al. 2006), CGCM3.1(T47) (Scinocca et al. 2008), and UKMO-HadCM3 (Pope et al. 2000) models. Outputs were taken from the three GCMs run for the B1, A1B, and A2 greenhouse-gas emissions scenarios (Nakicenovic et al. 2000) as part of the World Climate Research Programme's (WCRP's) Coupled Model Intercomparison Project, phase 3 (CMIP3) (Meehl et al. 2007). The B1 emission scenario assumes relatively sustainable global development. In contrast, the A1B and A2 scenarios focus on economic development. The A1B

scenario assumed globally integrated development whereas the A2 scenario portrays more heterogeneous, regional development. Consequently, the B1 scenario results in the lowest emissions of the three scenarios, and the A2 results in the highest. The AOGCM data were obtained from the CMIP3 multi-model dataset archived by the Program for Climate Model Diagnosis and Intercomparison (PCMDI; http://www-pcmdi.llnl.gov/ipcc/info_for_analysts.php).

Monthly mean temperature, total precipitation, and cloud cover anomalies for the years 2001-2099 and 1961-1990 30-year mean base period were calculated for each of the six GCM-emissions scenario combinations (see http://www-pcmdi.llnl.gov/ipcc/time_correspondence_summary.htm). Temperature anomalies were calculated as differences (monthly value minus 1961-1990 30-year mean base period value) and precipitation and cloud cover anomalies were calculated as ratios (monthly value divided by 1961-1990 30-year mean base period value). The future climate anomalies were downscaled to each study area grid cell using geographic-distance-weighted bilinear interpolation. The interpolated anomalies for each month from 2001-2099 were applied to the interpolated 1961-1990 30-year monthly mean data for each grid cell.

Vegetation-Change Projections

Two different models were used to simulate climate-driven vegetation changes across each of the three study areas (Shafer et al. In-Press). LPJ (Lund-Potsdam-Jena, ver. January 2004), a dynamic global vegetation model (Sitch et al. 2003), was used to simulate dominant vegetation types. LPJ-GUESS (ver. 030124, Smith et al. 2001), an ecosystem model, was used to simulate vegetation changes for a smaller domain surrounding Fort Benning and Fort Hood. Both models simulate vegetation in the form of plant functional types (PFTs) such as grasses, broad-leaved conifers, or deciduous trees. The PFTs can be combined to represent major biomes and habitat types (e.g., forest, grassland). The models were run at a monthly time-step to simulate the physiological response of vegetation to changes in temperature, precipitation, and increased atmospheric CO₂ concentrations.

The two vegetation models make different assumptions and produce different outputs. LPJ simulates an average individual PFT for each grid cell. In contrast, LPJ-GUESS simulates the average individual for multiple PFT cohorts. LPJ-GUESS was used to evaluate simulated changes in the size and age of average individuals of a given PFT through time. The PFT parameters of both models were adjusted to better represent the age and relative fire resistance of the major plant taxa at each site. When detailed physiological and life history data were not available, we used data for related species or species with similar physiological characteristics. For all of the vegetation simulations, a spin-up period of 800 years was used to allow carbon pools to equilibrate and vegetation to be established from bare ground. After the spin-up period, vegetation changes were simulated for an additional 199 years using the monthly historic climate data described above for the years 1901-2000 and projected future climate data for the years 2001-2099. We used projected global mean annual atmospheric carbon dioxide (CO₂) concentrations of 549 ppm for the B1 scenario, 717 ppm for the A1B scenario, and 856 ppm for the A2 scenario by the year 2100 taken from the Integrated Science Assessment Model reference case simulations (Prentice et al. 2001). All vegetation simulations described above were prepared by Sarah Shafer, USGS.

The Red-cockaded Woodpecker Case Study

Multi-Scale Habitat Model

The following procedures were used to evaluate the potential for ensemble techniques to improve model accuracy and performance of finer scale habitat models. Habitat models were constructed for the population of Red-cockaded Woodpeckers at Fort Benning. Woodpeckers may select habitat based on both the attributes of potential nest locations and attributes of much larger foraging areas. Therefore, models were built with data collected at two corresponding spatial scales. To address the issue of variability in modeling approaches, models were also constructed using three different modeling techniques. The accuracies of predictions made with single models to those made with various model ensembles were compared.

All models were based on several simple attributes of forest stands. These attributes were derived from forest-inventory data collected in 2003 at Fort Benning. Although more recent data exist, the 2003 dataset was the most complete dataset available. Variables selected were known to be important to RCWs for both foraging and nesting, including age of stands, basal area of pine and hardwood stems, and functional type (conifer/non-conifer) as explanatory variables in our model (Table 1). These data were mapped to grids with 50-m by 50-m cells using ESRI ArcMap 9.3.

A subset of active RCW clusters at Fort Benning has been continuously monitored for breeding activity and demographics since 1996 (Doresky et al. 2001). As a component of this monitoring, the locations of each active and inactive cluster as well as the locations of all nest trees within each cluster have been mapped. Presences and absences were defined at two different spatial scales to build a “nesting-scale” model and a “foraging-scale model.” The geographic center of known RCW foraging territories was used as the nest location for each group. A 109-ha buffer (radius = 589.2 m) was added around each cluster center to represent the average size of RCW foraging areas at Fort Benning. These 109-ha regions served as samples of presences at the foraging scale. Two hundred and seventy-five presences were sampled at each scale. To measure attributes of areas where RCWs did not nest, 800 random points were generated within the Fort Benning boundary. Each random point was then buffered with a 109-ha buffer and any random point where the buffer overlapped with the foraging area of an active cluster was deleted, resulting in 303 random point locations (Figure 6). Because the locations of all RCW clusters are known at Fort Benning, these points can be considered true absences.

Table 1. Environmental variables used in constructing Red-cockaded Woodpecker habitat probability models. Data are derived from forest inventory data collected in 2003 at Fort Benning, Georgia, USA.

Variable name	Definition	Type	Range of values
Age(30)	Average age* of trees within stand is greater than or equal to 30, or less than 30 years old	Categorical	≥ 30 < 30
Age(60)	Average age* of trees within stand is greater than or equal to 60, or less than 60 years old	Categorical	≥ 60 < 60
Conifer	The functional type of the dominant species in the stand	Categorical	Conifer Hardwood
BA pine	Total basal area (ft ²) of pine stems within the stand	Continuous	0-210
BA hardwood	Total basal area (ft ²) of hardwood stems within the stand	Continuous	0-170
BA total	The sum of BA pine and BA hardwood	Continuous	0-220
Percent pine	Percentage of the total basal area in pine	Continuous	0-100

*Age parameters were adjusted to reflect current age of stands.

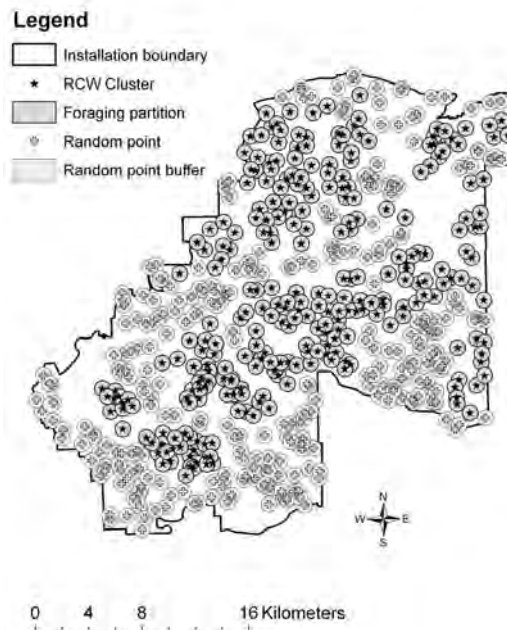


Figure 6. Red-cockaded woodpecker nest (presence) and random point (absence) locations at Fort Benning, GA. Points represent the geographic center of woodpecker clusters. Circular buffers around points represent the average 109 ha foraging area maintained by RCW groups.

GridSampler (Zerger 2004) was used to measure each of the predictor variables for each presence and absence observation at the nesting scale. A modified version of GridSampler (renamed CircleSampler) was used to sample the 109-ha foraging areas. The modified version allows the user to specify a radius length and then samples all pixels that fall within that radius around the central point (nest location). Within each foraging area, the percentage of the area with average tree age ≥ 30 , average tree age ≥ 60 , and for which the dominant tree type was conifer as calculated, as well as the average value for the basal area of pine, basal area of hardwood, total basal area, and the percent of the total basal area in pine.

Three modeling approaches were selected (Random Forest predictors, logistic regression, and generalized additive models) to build predictive models of habitat probability for RCWs at Fort Benning. These three approaches span a range of simple to more complex model structures and from basic statistical to machine learning approaches.

Random Forest predictors are a model averaging approach (Breiman 2001). A “random forest” is a statistical combination of multiple classification or regression trees in which each tree is constructed using a subset of the data and a subset of available predictor variables. Random Forests have been used to build species occupancy models and have performed as well as or better than other methods such as generalized linear models (Garzón et al. 2006, Lawler et al. 2006a, Cutler et al. 2007). Habitat models were built with the R statistical package *randomForest* (Liaw and Wiener 2002). Each model was based on 1000 trees. Each tree was generated using 2 randomly selected predictor variables as candidates for each split in the tree and a random selection of two-thirds of the training data.

Logistic regression models were built using a combined forward and backward stepwise selection procedure in which variable inclusion was determined using Akaike’s information

criteria (Chambers and Hastie 1991). All models were constructed using the `glm` function in R with an assumed binomial error distribution.

Generalized additive models (GAMs) that used logistic regression and cubic splines were used to model the relationships between categorical and continuous predictor variables, respectively, and RCW presence/absence. Model selection for categorical variables was based on a backward stepwise selection procedure. Cubic splines were fit to continuous predictors using penalized least squares regression (Hastie and Tibshirani 1990). Variable selection for continuous predictors was based on an estimated p-value generated by shrinking the smoothness parameter of each spline to zero in order to estimate its contribution to the model fit (Wood and Augustin 2002). The `gam` function from the MGCV package in R was used to fit all GAMs using an assumed binomial error distribution.

For each modeling approach, a nesting-habitat model and a foraging-habitat model were built using data from the two distinct spatial scales as well as a multi-scale model using both nesting- and foraging-scale data in a single model (Figure 7). Several sets of ensemble models were also created. Scale ensembles (Figure 7) were created by combining predictions from the nesting-habitat model and the foraging-habitat model for each approach. Ensembles of the different modeling approaches were created by combining predictions from all three types of models, applied to each of the three model scales (nest, forage, and multi-scale models; Figure 7). The predictions from each model were combined through both weighted (by AUC) and unweighted (committee) averaging (Marmion et al. 2009).

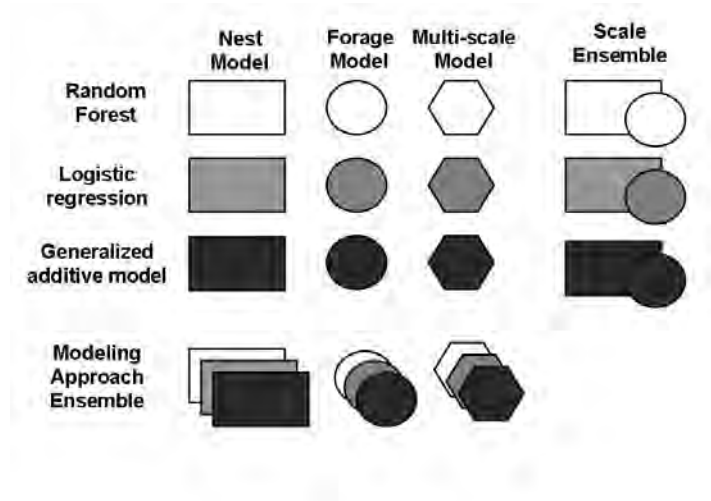


Figure 7. Schematic illustrating the various models built for each modeling approach and scale. Each modeling approach is represented by a different color and the different scales are represented by different shapes. The bottom row and far right column illustrate which models were combined for each ensemble.

Two different test data sets were used to assess model performance and accuracy. The Fort Benning data were divided into a training set and a test set by randomly withholding one-

third of presences and one-third of absences (90 points each) to serve as test data. Although validating models using reserved (semi-independent) data sets is common, these data may be spatially or temporally autocorrelated with the training data, resulting in overly optimistic estimates of model accuracy (Araújo et al. 2005). Because these test data could be spatially autocorrelated with the training data used to build the models, the models were also tested using completely independent data from Fort Bragg, North Carolina, USA. Fort Bragg has approximately 450 active Red-cockaded Woodpecker clusters within the 65,033-ha installation. Three hundred forty two absence points were generated and data collected using the same techniques described for Fort Benning. The models built with Fort Benning data were used to predict RCW presence and absence at Fort Bragg.

Model performance was evaluated based on the test data using two criteria. First, receiver operating characteristic (ROC) curves were generated for each model using the PresenceAbsence package in R (Freeman and Moisen 2008b) and the area under the curve (AUC) was calculated as an indicator of model performance. Second, thresholds for classifying presences and absences from the predicted probabilities for each model were selected using the MaxKappa function in PresenceAbsence. The MaxKappa function was selected because it has been shown to have low bias in predicting species presences and also maximizes kappa (Freeman and Moisen 2008a). Model accuracy was assessed by calculating the percentage of test data (both presences and absences) correctly predicted by each model.

Population Simulations in HexSim

The following procedures were used to explore the potential effects of multiple stressors on the Red-cockaded Woodpecker population at Fort Benning. Specifically, the Hexsim modeling platform was used to build a population model that simulated the potential effects of land-use change, climate change, and aging loblolly and longleaf pine stands.

Each simulation used the same baseline habitat map (Figure 8). This map was generated using the multi-scale RCW habitat model discussed above (*Multi-scale Habitat Model* section). However, the baseline habitat map was constructed with a simplified data set consisting of age 30, age 60, and conifer maps (Table 1). This simplified habitat map was slightly less accurate (presence: 87%, absence: 81%). However, unlike the complex habitat model, all parameters in the simplified model could be projected under future conditions. This baseline habitat map was the foundation for all further habitat maps generated under the scenarios described below.

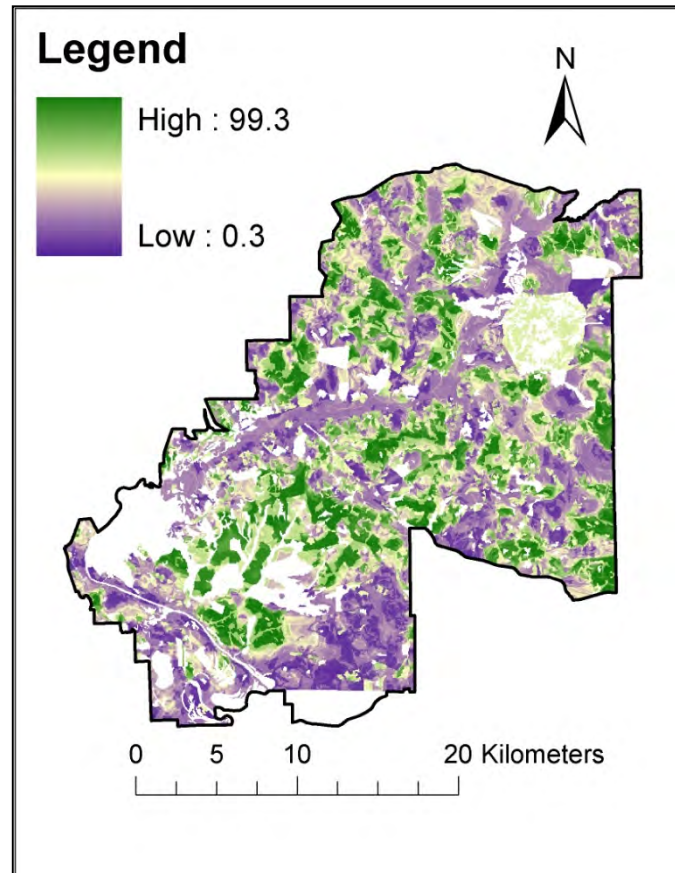


Figure 8. Baseline habitat map for all Red-cockaded woodpecker HexSim simulations. Green indicates high quality habitat, while purple indicates low quality habitat. This map was generated using tree age (≥ 30 and ≥ 60) and tree species (conifer or not conifer).

Simulating Land-use Change. To simulate the effect of recent troop increases on Fort Benning, a map of proposed training ranges was used as a baseline of future development (Figure 9). These ranges do not represent actual training range development on post, but are useful to estimate the land area that might change due to Transformation projects. The ranges shown in Figure 12 were added to the baseline landscape (habitat map shown in Figure 8) by downgrading the habitat quality within each range according to the estimated impact of each range on RCW habitat (Figure 9; M. Barron, personal communication). This new habitat map with the Transformation ranges in place, hereafter “baseline development map” (Figure 10), was used to generate future development scenarios representing different land use decisions. In the development scenarios, three or four new ranges were added every ten years during the 100-year simulations. Thus, the baseline development map reflects the addition of approximately 13,000 troops, while each new development scenario reflects development of new training ranges necessary for an additional 13,000 troops over 100 years, for a total of 26,000 new troops under each development scenario. The area of land in new training ranges in each development scenario doubles the current area developed due to the Transformation. Importantly, a static, non-aging forest was assumed in these scenarios.

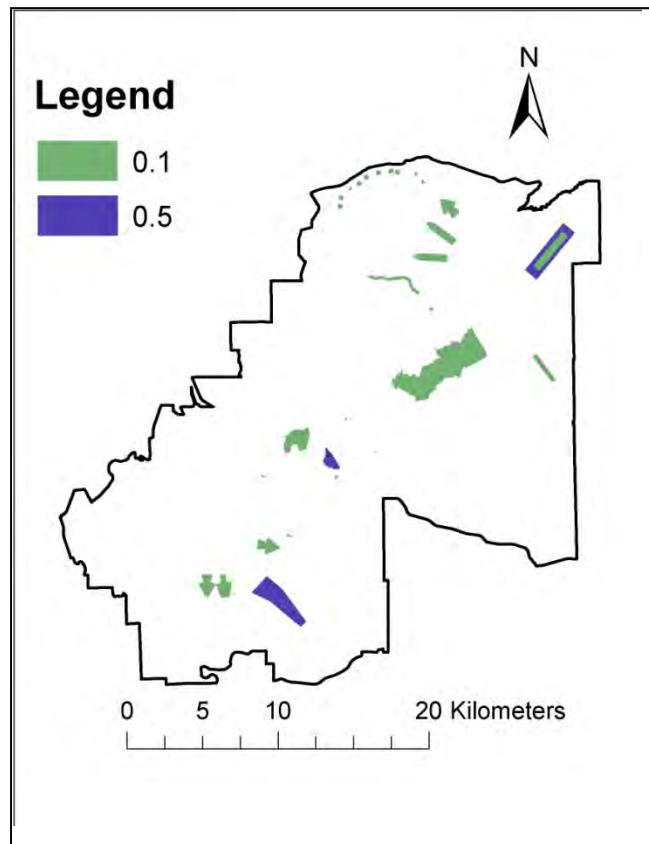


Figure 9. Proposed training ranges reflecting the Transformation at Fort Benning. Although this map does not represent the current plans for range development on post, it was the most current available at the time of scenario development and implementation in HexSim. This map formed the baseline range development scenario from which all other range development scenarios were generated. Values indicate the effect of each range on RCW habitat as the habitat quality was multiplied by the value shown to downgrade the habitat. Map provided by M. Barron.

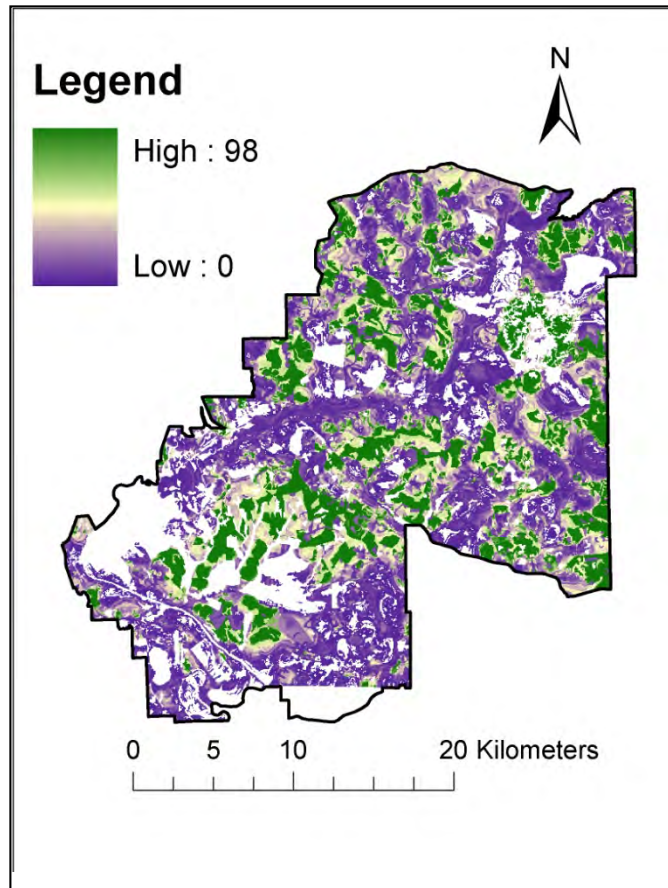


Figure 10. Baseline development habitat map for Red-cockaded woodpeckers (RCW) using proposed training ranges at Fort Benning, GA. Each range was placed on the landscape and habitat within the range was downgraded based on estimated impact of the range on Red-cockaded woodpecker habitat quality. Green areas indicate high quality RCW habitat, while purple areas indicate low quality RCW habitat.

Three classes of future land-use change scenarios (Conservation, Convenience, and Worst) were developed with three development scenarios in each class. Each development scenario within each class differs from the others in physical location of each range, but the rules for placing ranges are the same within a class. Ranges were randomized prior to addition to the landscape such that the order of placement varied in each scenario. For each class, ranges were placed randomly within designated development regions (which varied by class) and habitat quality was downgraded based on the estimated impact of each range on RCW habitat as in the baseline development map. Ranges were randomly assigned to the year of placement (in 10 year increments) and then placed on the landscape. Each range was first centered on the assigned random point and then adjusted as necessary to avoid overlapping any existing training range or duded area. If a range was too large to fit in the assigned area, the range was moved to the next random point in the map, in order of random point generation. Three different point and range order randomizations were generated to create three different land use scenarios within each class.

Once all ranges were in place, new habitat maps were created by multiplying the habitat quality of each pixel in the baseline development map by the value of each range shown in Figure 9. Maps were generated for each 10 year increment, reflecting that decade's new ranges, plus any previous decade's ranges so that by year 2090 all ranges were in place on the final map for each scenario.

Conservation scenarios. Conservation land-use change scenarios were designed to minimize the placement of ranges in RCW habitat. Habitat below the probability threshold calculated using Max Kappa (see *Multi-scale Habitat Model* section) was designated as “non-RCW habitat” and available for development. A new shapefile of non-RCW habitat was created in ArcMap 9.3 (ESRI 2008) and 1000 random points were generated within this shapefile. Three different randomizations were generated, resulting in different spatial arrangements and year of placement for each Conservation scenario (Figures 11 & 12).

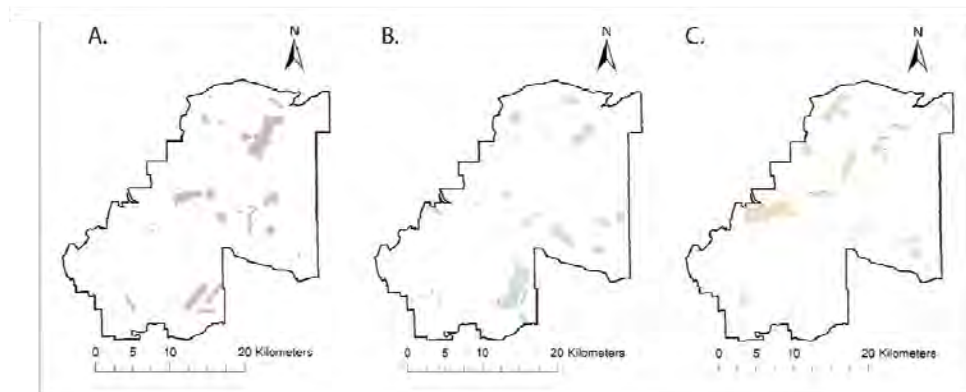


Figure 11. Placement of new training ranges under the three Conservation scenarios for Red-cockaded woodpecker (RCW) simulations. Ranges were placed in non-RCW habitat to simulate land use decisions designed to minimize development effects on RCW habitat.

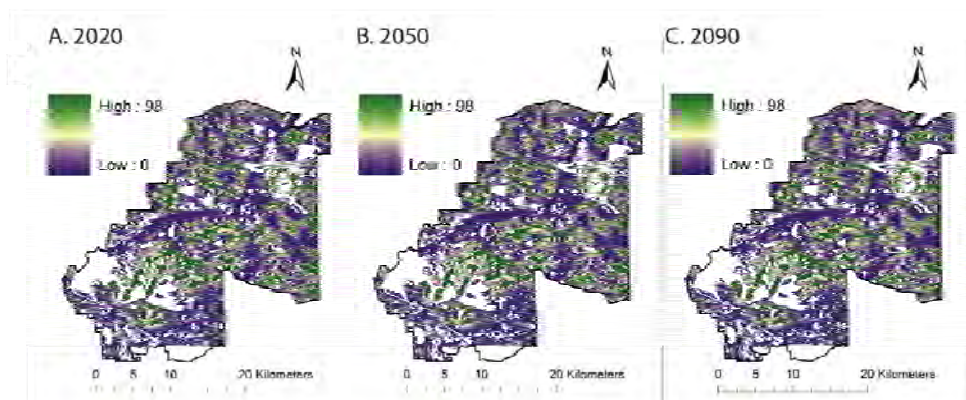


Figure 12. Conservation (scenario 1) habitat maps for Red-cockaded woodpecker (RCW) land use change simulations. Ranges were added to the landscape at 10-year intervals over 100 years. Years 2020 (A), 2050 (B), and 2090 (C) are shown. Green colors indicate high quality habitat, while purple colors indicate low quality habitat.

Convenience scenarios. Convenience land-use change scenarios were designed to simulate land use change based on location of roads on the installation. Using ArcMap 9.3 (ESRI 2008), current roads on Fort Benning were buffered with a 400 m buffer. As in the Conservation scenarios, 1000 random points were generated within this buffer and new ranges placed using these random points. Thus, each new range in the Convenience scenarios was placed within 400 m of an existing road on the installation. As much of the installation is near a road, the convenience scenarios represented nearly random placement of training ranges at Fort Benning. Three different randomizations were generated, resulting in different spatial arrangements and year of placement for each Convenience scenario (Figures 13 & 14).

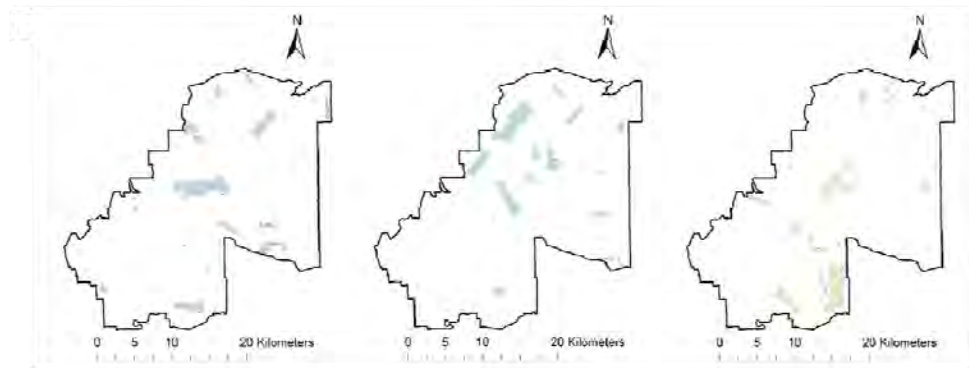


Figure 13. Placement of new training ranges under the three Convenience scenarios for Red-cockaded woodpecker (RCW) simulations. Ranges were placed within 400 meters of existing roads.

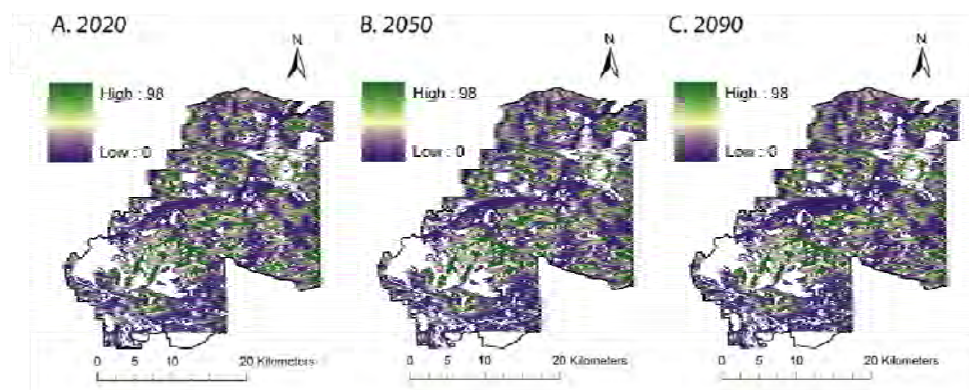


Figure 14. Convenience (scenario 1) habitat maps for Red-cockaded woodpecker (RCW) land use change simulations. Ranges were added to the landscape at 10-year intervals over 100 years. Years 2020 (A), 2050 (B), and 2090 (C) are shown. Green colors indicate high quality habitat, while purple colors indicate low quality habitat.

Worst-case scenarios. Worst-case land-use change scenarios were designed to simulate training range development only in areas of RCW habitat. Habitat above the probability threshold was designated as “RCW habitat” and available for development. A new shapefile of

RCW habitat was created in ArcMap 9.3 (ESRI 2008) and 1000 random points were generated within this shapefile. As in the Conservation scenarios, ranges were randomly placed in the landscape using these random points. Three different randomizations were generated, resulting in different spatial arrangements and year of placement for each Worst-case scenario (Figures 15 & 16).

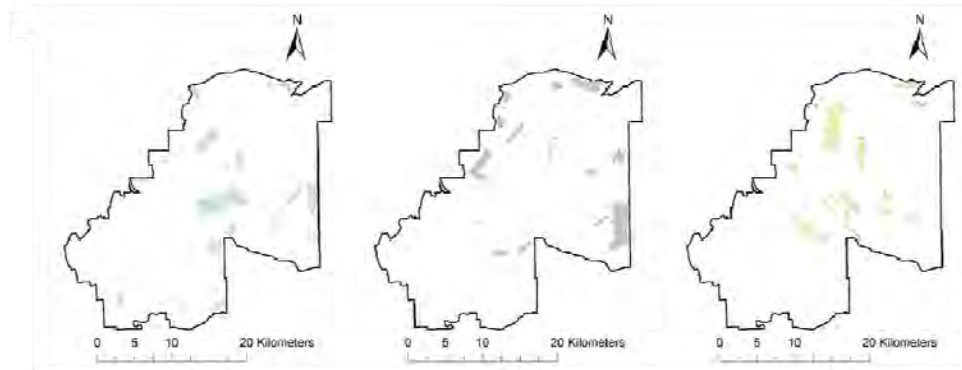


Figure 15. Placement of new training ranges under the three Worst-case scenarios for Red-cockaded woodpecker (RCW) simulations. Ranges were placed in RCW habitat to simulate land use decisions designed to maximize development effects on RCW habitat.

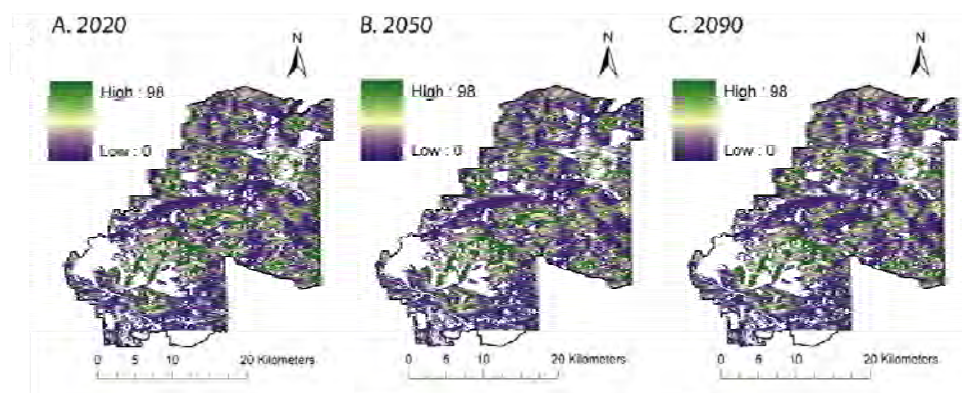


Figure 16. Worst-case (scenario 1) habitat maps for Red-cockaded woodpecker (RCW) land use change simulations. Ranges were added to the landscape at 10-year intervals over 100 years. Years 2020 (A), 2050 (B), and 2090 (C) are shown. Green colors indicate high quality habitat, while purple colors indicate low quality habitat.

Simulating Climate Change. The potential effects of climate change on RCW at Fort Benning were assessed in two ways. First, the effect of future climate change on vegetation was assessed as described in the section above titled *Vegetation-Change Projections*, using future climate data generated by three coupled atmosphere-ocean general circulation models (AOGCMs), CCSM3 (Collins et al. 2006)CGCM3.1(T47) (Scinocca et al. 2008), and UKMO-HadCM3 (Pope et al. 2000). Because no changes in the dominant vegetation were projected for Fort Benning (see the

Results and Discussion section titled *Vegetation Change Projections*), the effects of future vegetation were not modeled in HexSim.

Second, future climate projections were used to model the potential effects of precipitation on RCW population persistence. An empirical relationship between RCW reproductive success and April precipitation was detected at Fort Benning (Figure 17). This relationship was used to project the effects of future climate (as described in section 2.2 *Climate-Change Projections*) on reproduction in RCW by incorporating total April precipitation (mm) for each year mapped to 50 m grids in HexSim. Within Hexsim, the average precipitation within a cluster territory was used to generate reproductive rates based on the relationship in Figure 20. Therefore, reproductive rates were directly tied to the amount of precipitation projected within each cluster's foraging partition.

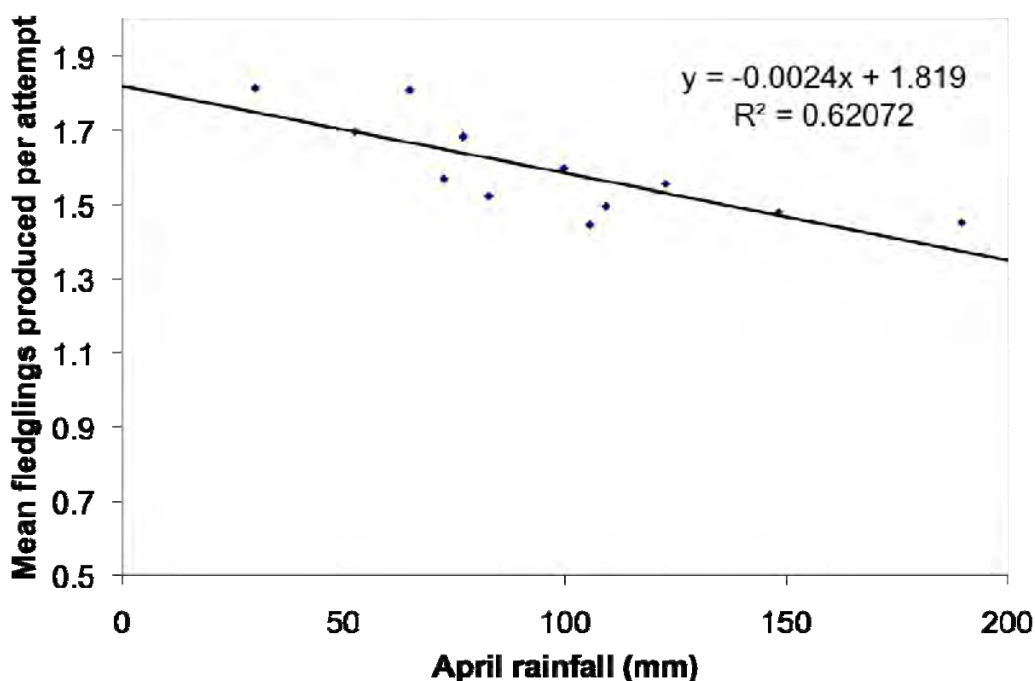


Figure 17. Correlation between April precipitation and Red-cockaded woodpecker (RCW) reproductive success at Fort Benning, GA. Reproductive success was lower in years with more precipitation.

Simulating Loblolly Decline. To simulate the effects of loblolly decline and restoration on RCW habitat and RCW population persistence, we created four scenarios with a dynamic (aging) forest (Table 2). All scenarios assumed that longleaf pine naturally senesced at 160 years of age and loblolly pine senesced early at 60 years of age. All scenarios also assume that mixed pine stands have enough longleaf pine to serve as habitat for RCWs until the stand reaches 160 years old. Given that some stands likely do not have enough longleaf to serve as habitat for that long, all of these scenarios can be seen as biased towards predicting more suitable habitat and larger RCW

populations. The four scenarios included a restoration scenario, a no-restoration scenario, a harvest scenario and a selective-harvest scenario (Table 2).

Table 2. Forest management scenarios simulating the effects of loblolly decline on Red-cockaded woodpecker (RCW) habitat. Each scenario assumed a dynamic aging forest.

Scenario name	Assumptions of scenario
Restoration	Declining loblolly stands (age 60) are harvested and replanted with longleaf.
No restoration	Declining loblolly stands are not replanted
Cut	Stands are harvested and replanted with longleaf. Stands are selected based on the number of trees in each 5-year age class. Age classes with the most trees are selected. Only stands >40 years old are cut. A maximum of 60 stands are cut in each 5-year period for the first 50 years in the simulation (no stands are cut after the first 50 years). Preference is given to loblolly stands for cutting.
Interplant	Individual trees within a stand are harvested and the gaps replanted with longleaf. The stands are selected based on the same criteria for the “cut” scenario. This scenario creates stands with two age classes.

Restoration. The restoration scenario assumed that as loblolly stands reach the age of 60 and senesce, they will be replanted with longleaf pine. The Nature Conservancy has worked with the base over the last 15 years or so to restore 11,000 hectares of longleaf pine forest by replanting stands of dying loblolly pine. If restoration were to continue at this rate it would be able to keep pace with the declining loblolly stands.

No-restoration. The no-restoration scenario assumes that dying loblolly stands will not be replanted. These unplanted stands are assumed to be incapable of acting as habitat for the length of the RCW population simulations.

Cut. The “harvest” scenario assumes that a number of stands will be harvested and replanted with longleaf.

Interplant. In this scenario, individual trees within a stand are harvested and the gaps replanted with longleaf.

Maps of tree ages were generated at 5-year intervals based on each of the four scenarios. These tree-age maps were then incorporated into the Random Forest model to generate new predicted habitat maps at 5-year intervals for each of the four management scenarios (Figures 18-21).

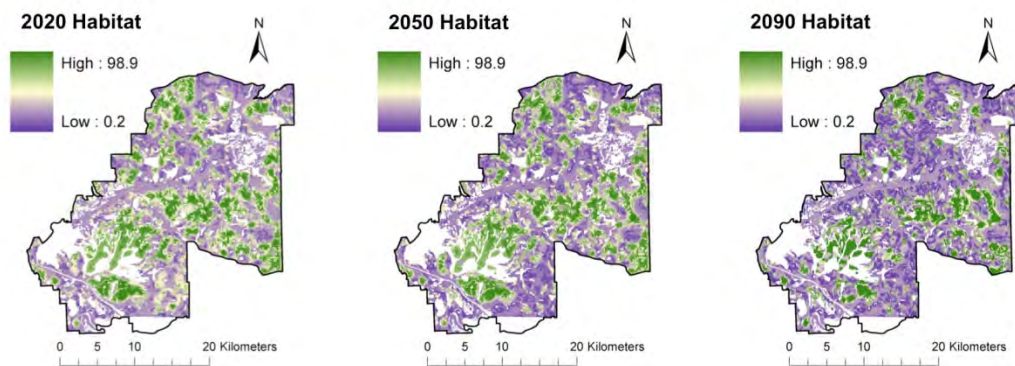


Figure 18. RCW habitat maps generated with a “No restoration” forest management scenario for the years 2020, 2050, and 2090.

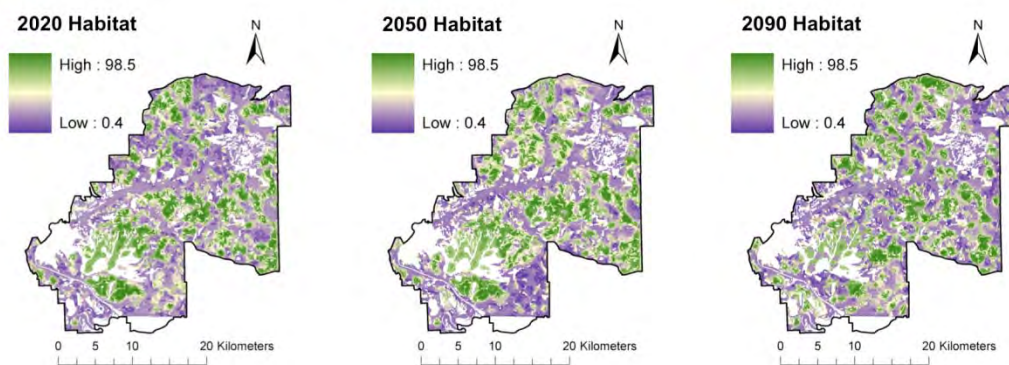


Figure 19. RCW habitat maps generated with a “Restoration” forest management scenario for the years 2020, 2050, and 2090.

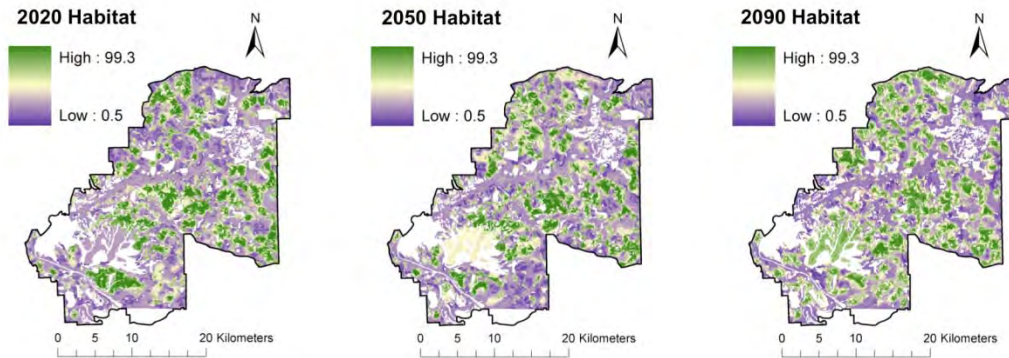


Figure 20. RCW habitat maps generated with a “Cut” forest management scenario for the years 2020, 2050, and 2090.

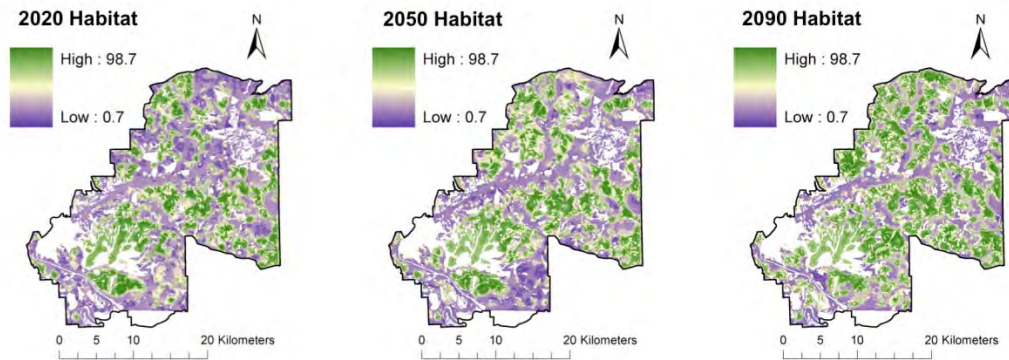


Figure 21. RCW habitat maps generated with an “Interplant” forest management scenario for the years 2020, 2050, and 2090.

HexSim Simulations. HexSim was parameterized using data from the literature (Walters et al. 1988, Walters 1990, DeLotelle and Epting 1992, LaBranche and Walters 1994, Engstrom and Sanders 1997, Conner et al. 1999, Daniels and Walters 2000, Doresky et al. 2001, Leonard. et al. 2003, Conner et al. 2004). Each simulation was allowed to stabilize (the “burn-in period”) for 50 years prior to introducing new landscapes or other data into the model. Baseline population trajectories were targeted to match current number of potential breeding pairs on the landscape under current conditions at Fort Benning. There are more rigorous and intensive ways to parameterize the HexSim model. Some of these methods are discussed in the section describing the Black-capped Vireo case study, below and others have been documented in the HexSim User’s Manual (Appendix B).

The Black-capped Vireo case study

To explore the effects of military training, climate-driven changes in the fire regime, and the control and cessation of control of cowbirds on Black-capped Vireo population three models were built and linked: a vireo-cowbird spatially explicit population model, a vireo habitat suitability model, and a vegetation growth and disturbance model. Figure 22 provides a flow chart for the model system highlighting the three models (rectangles), inputs and outputs, and entry points for training, cowbird management, and fire disturbance scenarios (ovals). The Vegetation Growth and Disturbance Model is a simulation model of future vegetation growth, military training impacts, and fire disturbance events. The model outputs maps of simulated vegetation for Fort Hood. These maps describe vegetation type, canopy heights, and edge density and are output at five-year intervals. The Vireo Habitat Suitability Model inputs data layers for vegetation type, canopy height, edge density, and soil depth and outputs maps of projected probability of vireo occurrence, a surrogate for habitat suitability. Habitat suitability maps are input into the Vireo-Cowbird Spatially Explicit Population Model. The individual-based population model follows the life cycle of each male individual in the population as they establish a territory, mate and reproduce, interact with cowbirds, undergo migration and survival, and disperse to form a new territory the following year. Habitat suitability data is incorporated into the population model when individuals disperse and establish territories. Outputs from the population model include annual population estimates for each series of projected vegetation maps.

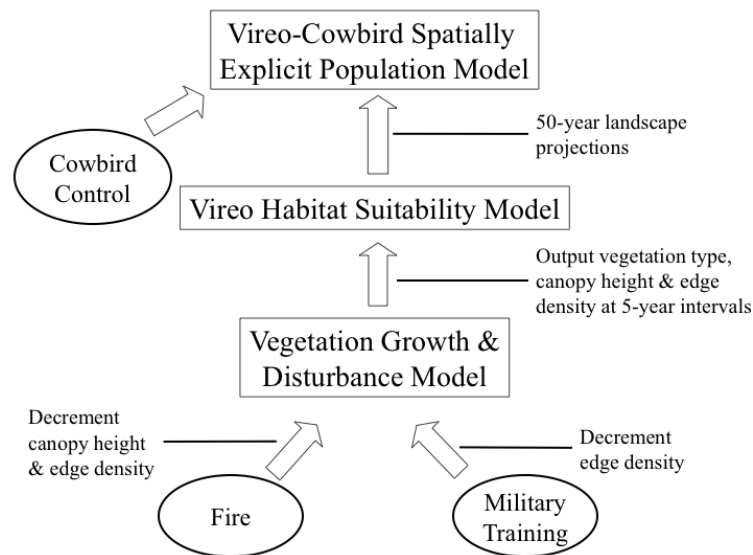


Figure 22. Modeling flow-chart for the Fort Hood Black-capped Vireo.

Vireo Habitat Model

As for the RCW, the first step in modeling Black-capped Vireo responses to multiple interacting stressors was to develop a habitat model that could be applied to both the current and future projected landscapes at Ft. Hood. The model uses information about vegetation composition and structure, soils, and landscape characteristics such as edge density to identify potential habitat for BCVI at Ft. Hood (Table 3, Figure 22).

Table 3. Response and predictor variables and data sources used in constructing a Black-capped Vireo Habitat Model for Fort Hood, TX.

Variable	Description	Original Format	Reference
VIREO	All areas identified as potential vireo habitat were visited in 2002 and 2003. Repeat visits on foot and playback were used to assess presence.	Point shapefile	(Cimprich and Kostecke 2006)
FXNAL	Manual delineation and classification of vegetation type based on aerial imagery and vegetation sampling data. Twenty-eight vegetation types identified, e.g., grassland, oak savanna, riparian forest.	Polygon (0.5 ha)	(Reemts and Teague 2007)
DEPTH	Soil depth to a restrictive layer extracted from the Soil Survey Geographic Database (SSURGO) produced by the US Department of Agriculture Natural Resource Conservation Service	Polygon (0.5 ha)	(USDA-NRCS 2007)
HEIGHT	Height of woody vegetation. Calculated with LiDAR return data from surveys conducted on Fort Hood in March 2009.	LAS files and LiDAR-derived DEM (2 m)	(Optimal Geomatics 2009)
COVER	Percent cover of woody vegetation. Percent of LiDAR returns measuring between 1 and 30 m in height.	LAS files and LiDAR-derived DEM (2 m)	(Optimal Geomatics 2009)
COVER2	Percent cover of woody vegetation less than 3 m tall. Percent of LiDAR returns measuring between 1 and 3 m in height.	LAS files and LiDAR-derived DEM (2 m)	(Optimal Geomatics 2009)
EDGE	Edge density. Total edge length within a grid cell divided by the area of the cell. Edges delineated from the 3 m resolution HEIGHT grid.	HEIGHT grid (3 m)	

A map of vegetation functional types across Fort Hood was generated in 2007 through manual delineation of vegetation patches from aerial imagery and patch classification based on vegetation sampling data (Reemts and Teague 2007). Digital orthophotos (0.35-m resolution) were taken in 2004. The resulting vegetation map includes 7693 polygons classified into 28 vegetation types. The minimum delineated patch size was ~0.5 ha. Therefore, we converted the polygon data into a 75-m resolution grid, assigning classes based on the center point of each grid cell in order to preserve spatial pattern of the polygon dataset (Bian and Butler 1999).

Shrublands suitable to vireos on Fort Hood are known to occur in areas with shallow soils and rock pavement (Bailey and Thompson 2007). Therefore, a map of soil depth (cm) to a restrictive layer was extracted from the US Department of Agriculture Natural Resource Conservation Service Soil Survey Geographic (SSURGO) database for Bell and Coryell Counties, Texas (USDA-NRCS 2007) using the Soil Data Viewer 5.2 tool (USDA-NRCS 2008) for ArcMap 9.2. These maps are based on soil surveys conducted throughout Bell and Coryell counties in 1977 and 1985, respectively. The soil map includes 1233 polygons with soil depths ranging from 28-201 centimeters. The minimum delineated patch size was also ~0.5 ha, and we again converted this dataset into a 75-m resolution grid as described above.

LiDAR data were acquired over Fort Hood from 21-25 March 2009. LiDAR is an active remote sensing technology for mapping three dimensional surfaces. A LiDAR device emits laser pulses at regular intervals as it is flown above the target surface. The LiDAR sensor then records both the intensity of light reflected by the surface and the time it takes for the pulses to return. Recording of multiple returns and delays from millions of laser pulses gives an estimate of surface complexity in three dimensions. Raw LiDAR data was processed and quality checked by the contractor against 950 test points to estimate a 95% confidence interval for the DEM of ± 0.47 meters (Optimal Geomatics 2009).

Maps of canopy height and two measures of woody cover across Fort Hood were generated using the original geo-corrected LiDAR point cloud files (LAS file format) and the FUSION software package (McGaughey 2009). The `canopymodel` function in FUSION extracted the mean elevation of returns within each 3 m grid cell and then subtracts the ground elevation from the DEM produced by the contractor to estimate canopy height. Height data was aggregated into raster grids at six spatial resolutions (10, 25, 100, 250, and 500 m) in ArcGIS 9.3 (ESRI 2009) by assigning the mean value of the smaller cells to the larger cell.

A 3-m resolution map of vegetation height converted into a binary shrub / no shrub map was used to calculate edge density. Edge density, the total distance of all edges in a grid cell divided by the area, was calculated in a GIS at six spatial resolutions by summing the total edge distance from the 3-m resolution map within the larger areas.

Maps of percent cover for woody vegetation were generated using the `cover` function in FUSION (McGaughey 2009). This function defines canopy cover as the number of points above a height threshold divided by the total number of points within a grid cell. Cover maps were created for all vegetation and for only vegetation with heights between 1 and 3 meters. The percentage of total returns within this range serves as a proxy measure of the percent cover of low-level woody vegetation used for nesting by vireos (Bailey and Thompson 2007). Both cover measures were calculated directly from the LiDAR point clouds at the six spatial resolutions.

Presence/absence data was converted into raster grids at the six resolutions by designating as a presence any grid cell containing a presence location, and designating remaining cells as absences. The assumption of absence is appropriate because the entire military installation was searched for vireos during the 2002-03 survey.

All habitat modeling algorithms and statistical analyses were run in the R statistical package v2.6.1 (R Development Core Team 2008). Habitat suitability models were calculated with `cforest` (Strobl et al. 2007), a variation of the Random Forest Predictors (RF) classification algorithm (Breiman 2001). RF is a non-parametric machine-learning tool notable for (1) requiring no *a priori* assumptions about the relationship between predictor and response variables, (2) the ability to model nonlinear relationships and interactions among variables, and (3) high prediction accuracy (Prasad et al. 2006). In direct comparisons, RF performed similarly or better than parametric methods such as linear discriminant analysis, generalized linear

regression, classification trees, artificial neural networks, and generalized additive models (Lawler et al. 2006b, Prasad et al. 2006, Cutler et al. 2007).

The `cforest` function (CF) of the ‘party’ package version 0.9-995 in R (Hothorn et al. 2010) is relatively new and not as widely applied as RF. It is also a non-parametric method for recursive partitioning that produces a ‘forest’ of classification trees with bootstrapped subsamples of a dataset, but CF differs from RF in its method for selecting whether a variable is chosen for a given split in each classification tree. CF selects variables based on a conditional inference statistical framework (Strobl et al. 2007). This approach eliminates the need for pruning trees and results in less bias towards predictor variables with many potential cut points (Strobl et al. 2007). CF is recommended when predictor variables include both categorical and continuous variables and when continuous variables differ in range (Strobl et al. 2007).

One third of grid cells were set aside in a test dataset to assess accuracy and to calculate additional test statistics. Due to the fine spatial resolution of the input data and large extent of the study area, not all presence or absence grid cells could be used to construct CF models. The number of randomly selected grid cells used in the modeling process varied with grid cell size, ranging from 1615 cells for the 500-m resolution model to 6600 cells for the 10-m resolution model. Prevalence, the ratio of presence to absence grid cells, has been shown to impact calculations of some performance metrics (Allouche et al. 2006). A ratio of one presence to two absences was maintained in all test and training datasets. Preliminary analyses suggested that metrics were sensitive to the composition of the training and test datasets. Therefore, 25 training and test datasets were randomly generated at each spatial scale and calculated performance metrics for each dataset (as in Santos et al. 2010). Statistical comparisons were not conducted because points were shared among datasets, violating assumptions of independence. Therefore, boxplots were used to summarize performance metrics for each model.

Because there are many different measures of model fit, each with its own advantages and limitations (Fitzgerald and Lees 1994, Stehman 1997, Manel et al. 2001, Allouche et al. 2006, Lobo et al. 2008), several measures were calculated. The CF models were used to generate the probability of presence for each grid cell in the test dataset. the `rocdemo.sca` and `AUC` functions from the ROC package v1.16 (Carey 2007) were used to generate a receiver operating characteristic (ROC) curve for each CF model (Hanley and McNeil 1982) and to calculate the area under the ROC to assess model fit (Hanley and McNeil 1982, Fielding and Bell 1997). The ROC curve was also used to select a cutoff value for classifying presences from predicted probabilities that minimized the difference between the rates of omission and commission error (Jiménez-Valverde and Lobo 2007). A confusion matrix constructed with classifications of habitat from each model and dataset using this cutoff value was used to calculate Cohen’s κ statistic for agreement between two raters (Fitzgerald and Lees 1994) using the `Kappa` function in the `vcd` package (Meyer et al. 2010). A κ statistic provides a measure of accuracy adjusted for prevalence as well as the relative performance of the classifier across classes (Fitzgerald and Lees 1994, Manel et al. 2001). In addition, the overall accuracy of the classification based on the confusion matrix was reported. The use of overall accuracy is generally discouraged as a single metric of model fit because it masks the relative rate of omission and commission error (Fielding and Bell 1997). However, cutoff values were selected to minimize differences between these errors, so the overall classification accuracy generally reflected both omission and commission error.

AUC, κ , and overall accuracy were used to select the spatial resolution at which models including only LiDAR-derived measures performed best. Two multi-scale models with presence/absence data summarized at 25-m resolution were also constructed: one including only

the 75-m resolution vegetation and soil data, and a second including all LiDAR-derived measures summarized at the best performing resolution and the 75-m resolution vegetation and soil data. Each model was also used to estimate variable importance as in RF: calculating the mean decrease in model accuracy resulting from the random permutation of each predictor variable within each tree (Breiman 2001, Hothorn et al. 2010).

Simulating Habitat Change

Project RC-1541 was originally designed to use the vegetation projections generated in by the LPJ and LPJ-Guess models (see the section titled *Vegetation-Change Projections*) as inputs to a Vireo Suitability model to explore the potential effects of climate change on Vireo habitat. However, these models generally indicated that 1) the mixture of vegetation types on the base would not change under the different climate-change scenarios, and 2) fire would likely determine the relative percent cover of trees, shrubs, and grasses on the base (see the Results and Discussion section titled *Vegetation-Change Projections*). While these results are informative, they could not be directly used in the modeling of vireo population responses to climate change in a spatially explicit way. Therefore, we developed a vegetation model for the base.

The Fort Hood Vegetation Growth and Disturbance Model (FHVM) was built to simulate habitat change in response to changes in military training activities and climate-driven changes in the fire regime. More specifically, the objectives for building the model were to: 1) create a data-driven, mechanistic model of vegetation disturbance for Fort Hood, and 2) use the model to characterize the potential impacts of future scenarios for increased military training and increased area burned on vireo habitat availability. The model draws upon both empirical data and expert opinion to parameterize modules for vegetation growth, military training impacts, and annual fire events. Modeled disturbance results in transitions between vegetation types including oak species important to vireos on Fort Hood. The model also tracks changes in vegetation height and edge density, structural measures most important to identifying vireo habitat. Maps of vegetation height, edge density, and habitat quality for the vireo are output from the model at five year increments.

The FHVM was written in the R programming language v2.8 (R Development Core Team 2008). All spatial inputs and outputs are fit to a 75-m raster grid comprising Fort Hood. The surrounding landscape and urban areas on Fort Hood are excluded from the model. The model contains three modules: 1) annual vegetation growth, 2) military training disturbance, and 3) fire disturbance.

Fort Hood Vegetation. Fort Hood is located at the intersection of the Edwards Plateau and the Crosstimbers and Southern Tallgrass Prairie ecoregions. Topography includes numerous flat-topped buttes with steep gullies and mesic bottomlands. The installation is covered by woodlands and upland forest (47%), grasslands (34%), and small amounts of shrublands and riparian forests (4% and 3%, respectively). Vegetation types on Fort Hood were characterized into sixteen classes derived from the Fort Hood Vegetation Map (Reemts and Teague 2007). These sixteen classes were selected to highlight the eight most important habitat types occupied by vireos. These eight vegetation types were modeled explicitly in the FHVM (Table 4). The eight vegetation types also captured key transitions between juniper encroached and non-encroached vegetation types. Juniper is often eliminated from encroached stands following frequent or severe fire (Fuhlendorf et al. 1996). Stands with juniper are less suitable to vireos (Grzybowski et al. 1994).

Table 4. Vegetation types on Fort Hood. Those whose growth and change over time are explicitly modeled are in the Fort Hood Vegetation Growth and Disturbance Model are noted with an asterisk.

Vegetation Type	Description
*Mixed Oak Shrubby Savanna	<i>Quercus fusiformis</i> , <i>Q. buckleyi</i> , <i>Q. sinuata</i> savanna
*Mixed Oak Shrubby Savanna-encroached	<i>Q. fusiformis</i> , <i>Q. buckleyi</i> , <i>Q. sinuata</i> var. <i>breviloba</i> savanna encroached by <i>J. ashei</i>
*Oak Savanna	Associations of <i>Q. fusiformis</i> , <i>Q. buckleyi</i> , <i>Q. stellata</i>
*Oak Savanna-encroached	Associations of <i>Q. fusiformis</i> , <i>Q. buckleyi</i> , <i>Q. stellata</i> encroached by <i>J. ashei</i>
*Shin Oak Juniper Woodland	<i>J. ashei</i> , <i>Q. sinuata</i> var. <i>breviloba</i>
*Shin Oak Shrubland	<i>Q. sinuata</i> var. <i>breviloba</i>
*Texas Oak Juniper Slope Forest	Various associations of <i>Quercus muehlenbergii</i> , <i>Q. buckleyi</i> , <i>Fraxinus texensis</i> , <i>J. Ashei</i> , <i>Acer grandidentatum</i> , <i>Juglans major</i>
*Texas Oak Juniper Slope Forest-burned	Various associations of <i>Q. muehlenbergii</i> , <i>Q. buckleyi</i> , <i>F. texensis</i> , <i>J. Ashei</i> , <i>A. grandidentatum</i> , <i>J. major</i> recently burned
Grassland	Includes all herbaceous grasslands, native and disturbed
Juniper Monoculture	<i>Juniperus ashei</i> stands
Mesquite Woodland	<i>Prosopis glandulosa</i> dominated woodlands
Mesquite Woodland-encroached	<i>P. glandulosa</i> encroached by <i>J. ashei</i>
Riparian Forest	Various associations of <i>Carya illinoensis</i> , <i>Ulmus crassifolia</i> , <i>Elymus virginicus</i> , <i>Quercus macrocarpa</i> , <i>Salix nigra</i>
Riparian Shrubland	<i>Cephalanthus occidentalis</i> , <i>Ampelopsis arborea</i>
Riparian Woodland	Associations of <i>Q. fusiformis</i> , <i>U. crassifolia</i> , <i>Celtis laevigata</i>
Sumac Shrubland	<i>Rhus lanceolata</i> , <i>Bacharis neglecta</i>

Vegetation Growth Module. Growth of shrubland vegetation following fire can be rapid as existing roots and biomass facilitate re-sprouting. This growth pattern differs from a characteristic logistic plant growth function, such as the Richard's equation (e.g. Damgaard et al. 2002) in which growth is initially slow. Because of this difference, growth of the eight targeted deciduous shrubland vegetation types was modeled using a Michaelis-Menton equation for asymptotic growth.

$$height = \left(\frac{V_{max} \cdot t}{K_m + t} \right)$$

The Michaelis-Menton equation had the added benefit of requiring only two parameters, one of which could be estimated directly from available data. K_m characterizes the initial growth rate and V_{max} is the maximum height that a vegetation type will achieve. Growth models for each of the eight vegetation types were parameterized using measurements of initial shrub growth following a mulching experiment on Fort Hood and estimates of maximum vegetation height extracted from the 2009 LiDAR dataset (see Vireo Habitat Model Section for details on the data). V_{max} for each vegetation type was made equivalent to the 95th percentile of height measurements for all 75-m grid cells of that type. K_m values were selected to optimize fit to the

max height and initial growth data using non-linear least squares regression (nls function in R). Vegetation biologists have observed that vegetation over shallow soils grow more slowly than similar vegetation over deeper soils (Reemts, pers. comm.). Therefore, V_{max} was initially estimated using only grid cells with soils less than 50 cm deep. Then, a second set of growth equations for deep soils were defined by halving those values for K_m . Table 5 lists the shallow-soil parameter values for each modeled vegetation type. Figure 23 plots growth over time on shallow and deep soils.

Table 5. Modeled vegetation types for the Fort Hood Vegetation Growth and Disturbance Model and parameter values for the Michaelis-Menton growth equation. Values were estimated for vegetation growing on shallow soils (< 50 cm deep).

Vegetation Type	V_{max}	K_m
Mixed Oak Shrubby Savanna	10.23	29.34
Mixed Oak Shrubby Savanna-encroached	9.04	25.54
Oak Savanna	9.83	28.04
Oak Savanna-encroached	9.63	27.40
Shin Oak Juniper Woodland	9.24	26.15
Shin Oak Shrubland	7.07	19.22
Texas Oak Juniper Slope Forest	13.37	39.41
Texas Oak Juniper Slope Forest-burned	8.48	23.74

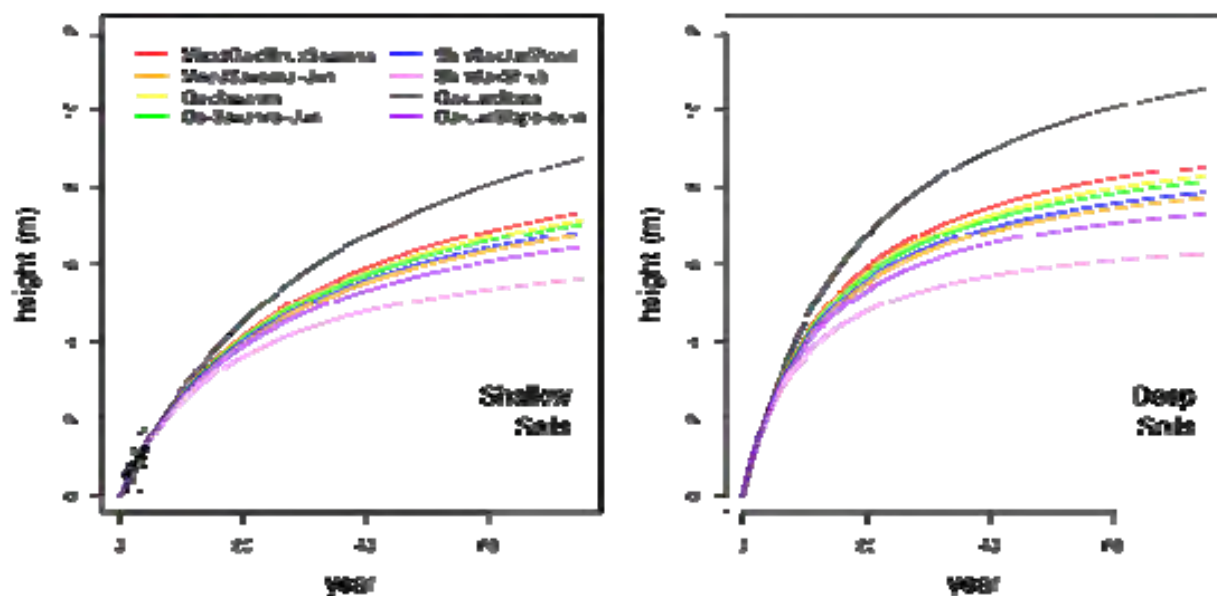


Figure 23. Modeled vegetation growth over time following a simulated fire disturbance. Growth rates are defined by a Michaelis-Menton growth equation.

In the absence of empirical data or a previous model, LiDAR-derived edge density estimates were used to parameterize a model of the changes in edge density following disturbance. A modified exponential growth equation was used because it allowed for an initially rapid increase in edge density post-fire followed by a slow decline toward canopy closure.

$$f(x) = (k_1 + k_2 \cdot x \cdot e^{k_3 x})$$

The k_1 parameter sets a y-intercept for the model, k_2 characterizes the initial rate of increase in edge density, and k_3 defines the rate for canopy closure (Table 6).

Table 6. Parameter values for the modified exponential growth equation used to model changes in edge density for the Fort Hood Vegetation Growth and Disturbance Model.

Soil Depth	K_1	K_2	K_3
Shallow	35.44	26.95	-0.078
Deep	-17.74	30.08	-0.090

A chronosequence approach using LiDAR-derived measures of edge density and known fire events from the past 15 years was used to parameterize the edge density growth model. Field ecologists at Fort Hood have mapped burned areas annually since 1994. The Fort Hood fire database includes 365 records of wildfires. Fire polygons were used to assign a time since disturbance to each grid cell impacted by recent fires. Grid cells were also grouped by soil depth (> and < 50 cm). Existing maps of designated Golden-cheeked Warbler habitat were used to characterize edge density in the mature oak-juniper woodlands preferred by warblers. Grid cells located in designated warbler habitat were assigned a time since disturbance of 50 years in shallow soils and 25 years in deep soils. The assigned time periods (50 or 25 years) were selected in consultation with the vegetation ecologist at Fort Hood as representative of the mean number of years required for canopy closure following disturbance. Finally, edge density values from recently burned areas and unburned warbler habitats were used to fit two models of edge density growth for all vegetation types (Figure 24) using non-linear least squares regression.

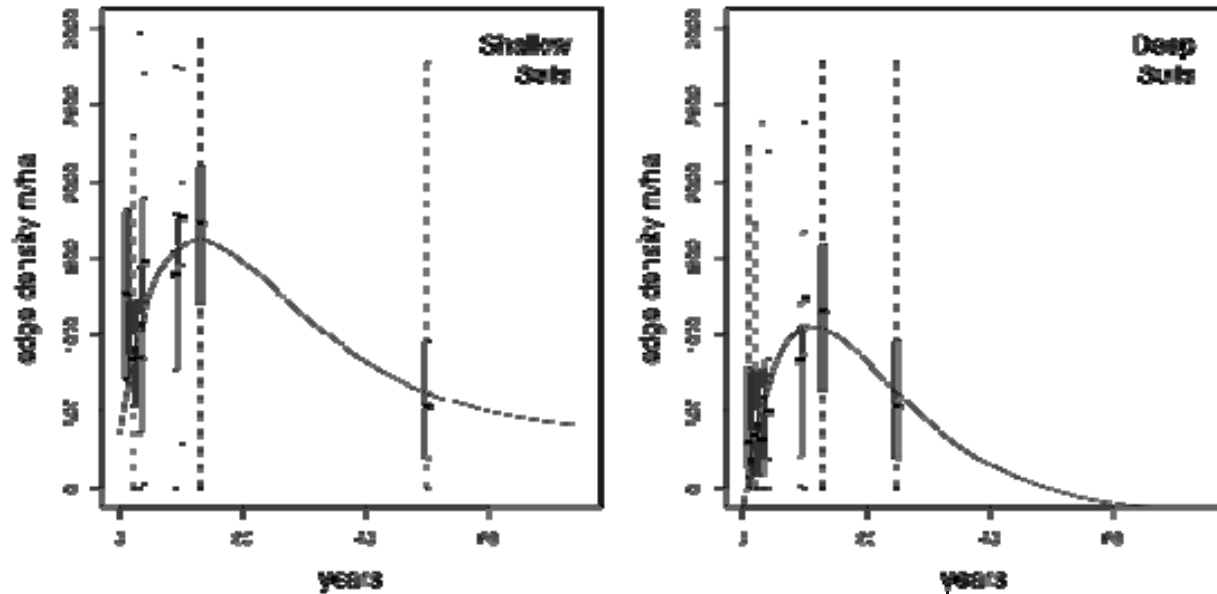


Figure 24. Modeled change in edge density following simulated fire disturbances. Boxplots are LiDAR-derived measures of edge density for mapped fires occurring since 1994.

The continuous growth curves for both height and edge density were used to produce growth tables with a discrete annual time step. The FHVM operates with an annual time step, so instantaneous growth rates were not required. Vegetation height growth in year $n+1$ depended upon how tall the vegetation already was in year n . Grid cells that were taller than the V_{max} did not continue to grow. Annual change in edge density increases immediately after a disturbance, but then declines as a grid cell moves toward canopy closure. Therefore, the model required that time since disturbance be determined for each grid cell upon initialization. Grid cells that had burned since 1994 were assigned the year in which they burned. Other grid cells were assumed to be moving toward canopy closure and assigned a year of last disturbance based on their current edge density value. A grid cell's year of last disturbance was reset by a fire event. Growth of recently disturbed grid cells with high edge densities was scaled to the maximum observed LiDAR-derived edge density on Fort Hood (~ 3000 m/ha). This was assumed to be a natural limit to edge density beyond which individual grid cells could not increase.

Military Training Module. The military training module simulates thinning due to heavy vehicle training. Training is limited to level areas (<8% grade) with forest and shrubland gaps large enough to allow vehicle maneuvers. Frequent training over a number of years is known to produce habitats characterized by patches of tall trees surrounded by shrubs, referred to as 'donut' habitat (Cimprich and Kostecke 2006). These areas have reduced woody cover, but tree size-class distributions are not significantly different from areas without tank training (Johnson 1982). The creation of this habitat was simulated in the FHVM by having training impact edge density but not vegetation height. This would occur in two ways. First, large scale thinning events would assign new edge density values to grid cells designated suitable for training. New edge densities were drawn randomly from an empirical distribution of LiDAR-derived edge

densities generated with grid cells currently designated as donut habitat (Cimprich and Kostecke 2006). Second, a switch was added to the annual growth module so that grid cells designated for training would randomly stop decreasing in edge density once having passed their maximum density on the density growth curve. These grid cells would stop changing edge density until another disturbance occurred. The pause in canopy closure would simulate the impacts of sustained vehicle traffic.

Fire Disturbance Module. The fire disturbance module simulates annual fire events throughout Fort Hood. The number and extent of fires were characterized based on wildfire records in the Fort Hood fire database. The annual number of wildfires at Fort Hood since 1994 followed a bi-modal distribution in which most years had relatively few fires and approximately two out of ten years had ten times as many fires. In the FHVM, the number of annual fires in the low-fire years was simulated by a random selection from a Poisson distribution ($\lambda = 5$). Whether a given year was a high fire-activity year was randomly determined by a draw from a binomial distribution ($p = 0.2$). In high fire years, the number of modeled fires was determined to be ten times the number randomly drawn from the Poisson distribution. Fire extents were randomly selected from a gamma distribution (shape = 7.304, scale = 0.503) fit to log-transformed estimates of fire extents from the Fort Hood fire database. Fire locations were also probabilistically assigned. Locations were drawn randomly from all grid cells in which the probability of selection was dependent upon (1) a grid cell's location within or outside of the Live Fire Area and (2) its relative fire return interval. Eighty percent of all fires in the Fort Hood fire database occurred within LF, so this proportion was used in the FHVM. Fire return intervals were defined by results of the LANDFIRE project (Keane et al. 2007), a simulation-based characterization of fire return intervals for vegetation types throughout the US. Vegetation types across Fort Hood have fire return intervals ranging from five (grasslands) to 1000 (riparian forest) years.

Simulated fires impacted both vegetation height and edge density. An area which had burned in a given year was assigned a new height and edge density randomly from empirical distributions of LiDAR-derived heights and edge densities. These distributions were composed of grid cells that had burned one or two years before the LiDAR data was collected in 2009 so as to provide an estimate of post-fire conditions. A randomly selected height that was taller than the current height of a burned grid cell was discarded until a new height was drawn that was equal to or less than the current height. Assigning post-fire values based on current distributions incorporated variation in fire intensity into simulations without modeling fire intensity explicitly. Maps of burned areas are output at five-year intervals.

Vireo Habitat Suitability Model. The FHVM uses a variation of the vireo habitat model described above (Vireo Habitat Model). A predictive model of vireo occurrence was generated using the Random Forest Predictors classification algorithm (Breiman 2001). Presence/absence data from the 2002-2003 surveys of Fort Hood were used to parameterize the model, and predictor variables included: the sixteen vegetation types listed in Table 4 and soil depth data and LiDAR-derived vegetation heights and edge density estimates listed in Table 3. All predictor variables were summarized at 75-m resolutions. The area under the receiver-operator curve (AUC) for this model was 0.859. The model was 77% accurate when predicting presence/absence in a test dataset of grid cells not used in model training. Cohen's Kappa (κ) statistic for the model was 0.527.

Sensitivity Analysis. A Monte Carlo Sensitivity Analysis (van Nes et al. 2002, Saltelli 2004) was performed to characterize the relative impact of model parameters on model output. Five hundred model runs were conducted with randomly selected inputs for each of ten model parameters. Parameter values were chosen randomly from a range spanning $\pm 20\%$ of the parameter's empirically-derived value. Growth function parameters were varied while maintaining the same proportional relationship between models for vegetation type and soil depth. Model output at time-step 30 was selected as the reference point for assessing parameter sensitivities. The relationship between parameter values and outputs were analyzed with multivariate linear regression in which model sensitivity was characterized by the coefficient for each parameter in the linear model.

Scenarios. Six future scenarios were generated to explore the potential impacts of military training, fire management, and climate-induced changes in fire disturbance on vireo habitat availability (Table 7). Scenarios were developed in consultation with wildlife biologists in the Natural Resources Management group at Fort Hood. Heavy vehicle training is currently most frequent on the west range, but managers are interested in clearing vegetation from the east range for additional training capacity. Therefore, scenarios include both of these possible futures. Two fire scenarios were explored. In the first, fire extents were increased by 30%. This is a relatively conservative projection of future change in area burned under climate change. Projections of future area burned in Canada given climate projections based on a tripling of current atmospheric CO₂ concentrations result in an average increase of 74-118% depending on the region (Flannigan et al. 2005). In the second fire scenario, fire frequency and size are held to historical levels and a policy of fire suppression is implemented outside of the LF area. The sixth scenario included both an expansion of training to the east range and a 30% increase in fire sizes.

Each scenario was run 30 times to incorporate some of the stochastic variation among model runs into the assessment of long-term impacts.

Table 7. Scenarios for the Fort Hood Vegetation Growth and Disturbance Model.

Scenario	Military Training	Fire
Baseline	None	Historic probabilities
Train West Range	West Range	Historic probabilities
Open East Range	West & East Range	Historic probabilities
Larger Fires	West Range	Increased fire extents
Fire Suppression	West Range	Historic probabilities, but no fires outside of LF
Train West & East Ranges and Larger Fires	West Range	Increased fire extents

Simulating Population Responses to Multiple Stressors

The HexSim spatially explicit individual-based population model was parameterized as a two-species (vireo-cowbird) population model to simulate the potential relative and cumulative effects of cowbird management, military training, and climate-induced changes in fire on vireo abundance. Impacts of military training and changes in fire were captured in the six scenarios run for the Fort Hood Vegetation Growth and Disturbance Model. The Vireo-Cowbird Population Model was run using these projected landscapes as inputs and scenarios with and without cowbird control.

The HexSim model was parameterized as a two-species, Vireo-Cowbird Population Model (VCPM). Figure 25 outlines traits modeled for each species and the model life cycle. Table 8 summarizes model parameters.

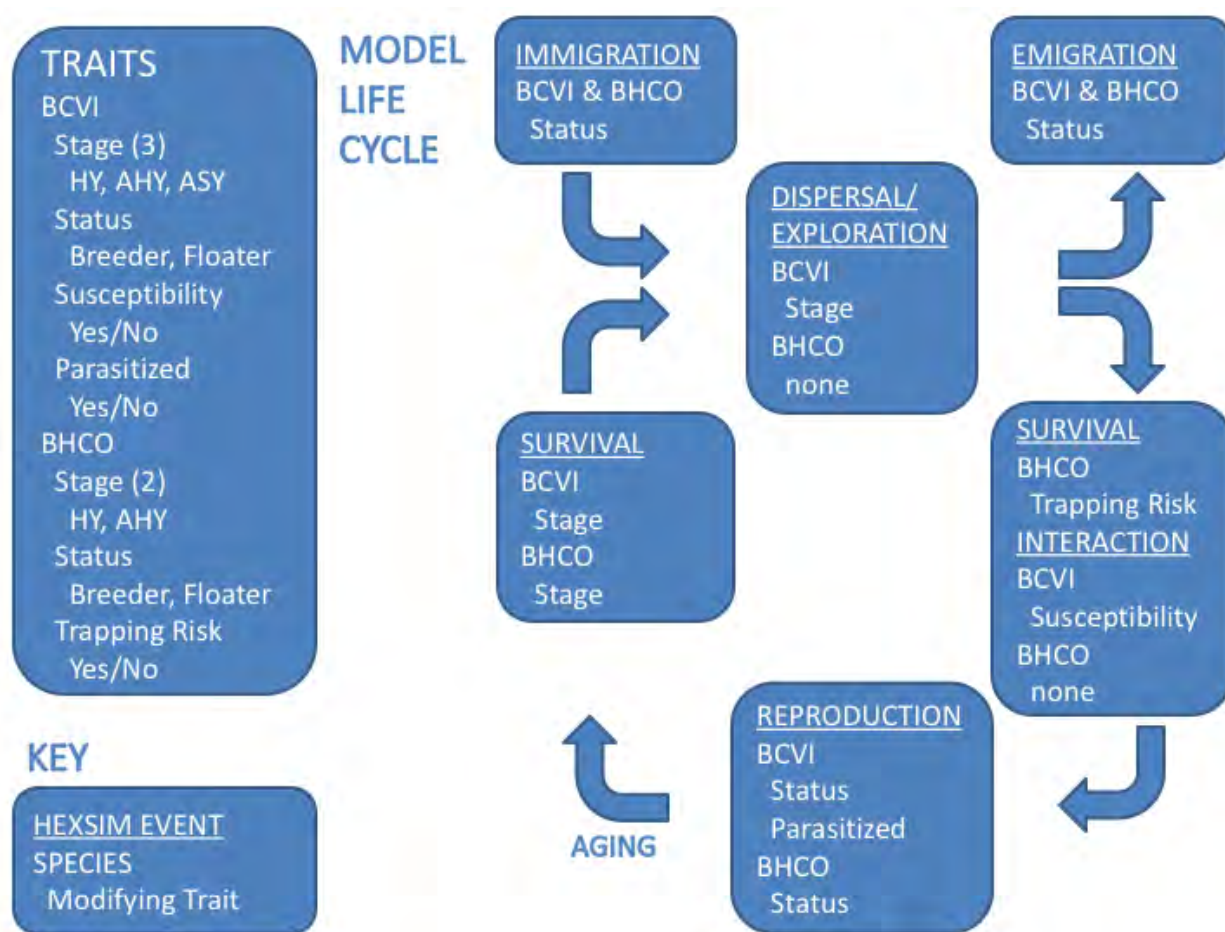


Figure 25. Species traits and modeled life cycle for a two-species population model in HexSim. BCVI = Black-capped Vireo, BHCO = Brown-headed Cowbird, HY = hatch year, AHY = after hatch year, ASY = after second year.

Table 8. Vireo-Cowbird Population Model parameters in HexSim.

Species	Category	ID#	Parameter	Data Source	Value
Vireo	Initialization	1	Initial Population Size	TNC-FHP (see text) reports	1500
		2	Maximum range area/span	TNC-FHP reports	9 ha / 800 m
	Range	3	Minimum range-eligible hexagon value	Vireo habitat model	1
		4	Minimum range resource	Vireo habitat model	8
		5	Resource target	TNC-FHP reports + Vireo habitat model	16
		6	Parasitism susceptibility	No data	85/15
		7	Maximum attempts	No data	6
		8	Max dispersal distance	(Kostecke and Cimprich 2008)	20 km
		9	Stopping criteria	Vireo habitat model	-
		10	Dispersal path percent autocorrelation	No data	75% 5 hex
		11	Trend period	No data	50 hex
		12	Maximum explored hexagons	Based on parameter 2	
	Reproduction	13	Maximum fecundity	TNC-FHP reports	8
		14	Growth matrix probabilities	TNC-FHP reports	2.7*
	Survival	15	HY survival	(Kostecke and Cimprich 2008)	0.44
		16	AHY survival	(Kostecke and Cimprich 2008)	0.6
	Immigration/ Emigration	17	Number of individuals/year	No data	200
		18	Floater survival	(Kostecke and Cimprich 2008)	0.4
Cowbird	Initialization	19	Initial Population Size	No data	6000
		20	Maximum range area/span	(Goguen and Mathews 2001)	28 ha / 800 m
	Range	21	Minimum range-eligible hexagon value	Cowbird habitat map	8
		22	Minimum range resource	Cowbird habitat map + (Goguen and Mathews 2001)	50
		23	Resource target	Cowbird habitat map + (Goguen and Mathews 2001)	252
		24	Trapping range	No data	3.5 km
		25	Shooting range	No data	100 m
	Dispersal/ Movement	26	Maximum attempts	No data	6
		27	Max dispersal distance	(Dolbeer 1982)	20 km
		28	Stopping criteria	No data	-
		29	% autocorrelation	No data	75%
		30	Trend period	No data	5 hex
	Reproduction	31	Maximum explored hexagons	Based on parameter 20	50 hex
		32	Maximum fecundity	(Payne 1976, Jackson and Roby 1992)	16
		33	Mean productivity	(Payne 1976, Jackson and Roby 1992)	10*
		34	SD productivity	(Payne 1976, Jackson and Roby 1992)	5*
	Survival	35	Trapping survival	No data	0.25
		36	Survival	(Ortega and Ortega 2009)	0.4
	Immigration/ Emigration	37	Number of individuals/year	No data	200
		38	Floater survival	(Kostecke and Cimprich 2008)	0.4

* *per territory*

HexSim is designed to model territorial species. Individuals in the VCPM establish territories based on resources distributed according to inputted spatial layers. Spatial inputs to HexSim are represented as hexagon grids (Figure 26). An annual time-step in the VCPM begins with aging and a survival event representing over-wintering survival (Figure 25). The subsequent dispersal/exploration event simulates the return from winter migration and formation of new territories. Individuals disperse from their location in the previous time step. A dispersal distance for each individual is selected from a uniform distribution bounded by minimum and maximum distance estimates. Dispersal paths are generated using a Markov Chain model with bias toward high quality habitats and away from low quality habitats. Dispersal ends when the individual reaches a patch of habitat of quality defined by stopping criteria. Following dispersal, individuals explore hexagons at their new location with the goal of establishing a territory. The *minimum required range resources* parameter in HexSim establishes how many resource units are required to form a territory. Dispersal/exploration cycles are repeated for individuals that fail to establish a territory immediately. If individuals are unable to accumulate enough resources to establish a territory after multiple attempts, they become non-breeding floaters. The inability of some individuals to form territories and reproduce introduces density-dependent limits on population growth. Immigrants are also added to the population midway through the dispersal/exploration sequence. A portion of all individuals unable to form territories is removed from the population as emigrants after the dispersal sequence.

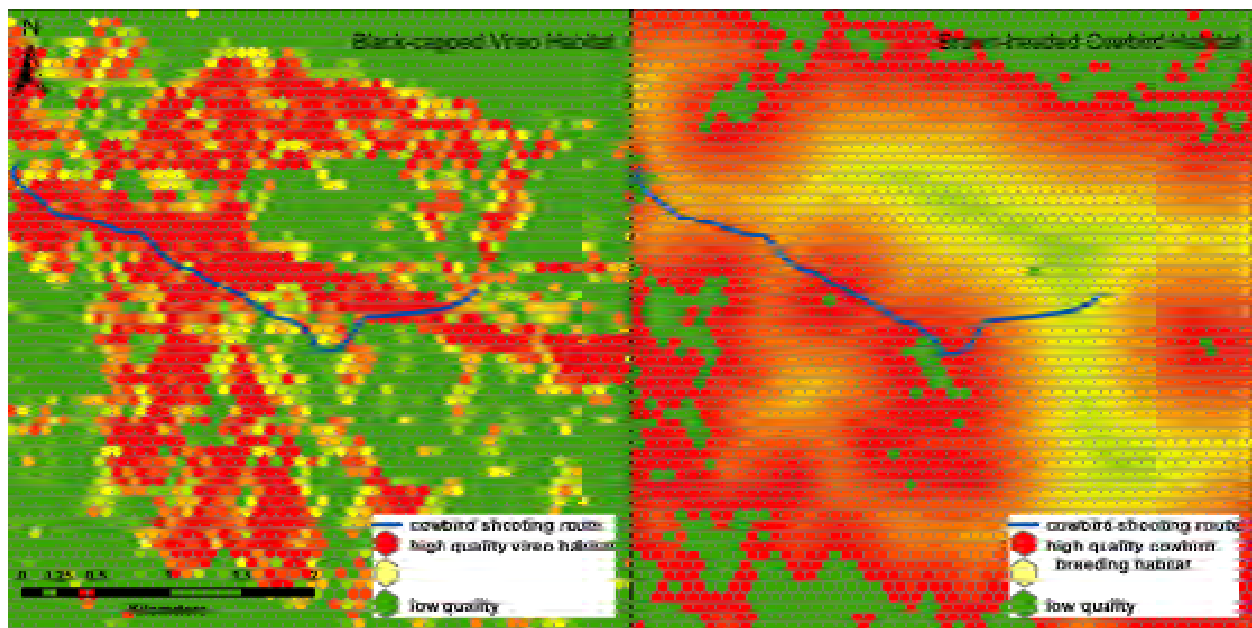


Figure 26. Spatial data in the Vireo-Cowbird Population Model is represented with hexagonal grids. Portions of the vireo and cowbird breeding habitat maps are shown. Cowbird shooting route added for frame of reference.

Traits in the VCPM are not passed from parents to offspring as genetic information, although HexSim can model genetic inheritance. Traits are age-dependent (Stage), based on accumulated resources (Status), or randomly assigned (Susceptibility, Figure 25). Traits

assigned to each individual in the population modify behaviors and vital rates. For example, survival parameters for the vireo (BCVI, Figure 25) are different for after hatch year (AHY) and after second year (ASY) individuals. Vireo reproduction and survival are also dependent on breeding status and stage. Similar traits influence cowbird behavior and vital rates. However, complexity was avoided in defining cowbird traits because relatively little is known about stage-specific vital rates for cowbirds on Fort Hood.

Brood parasitism is modeled using two vireo traits and an interaction event (Figure 25). The first trait is Susceptibility. All vireos are assigned as either susceptible or unsusceptible to parasitism when they are born at a ratio of 85/15. The interaction event identifies where vireo territories are overlapped by cowbird territories and assigns a value for the Parasitized trait based on the Susceptibility trait. An individual is designated Parasitized when 1) it is susceptible to parasitism and 2) its territory is overlapped by a cowbird territory. Individuals positive for the Parasitism trait have reduced productivity. The Susceptibility trait provides a single parameter for calibrating parasitism rates in the population.

Cowbird control is modeled with a spatial data layer and a single cowbird trait for trapping risk (Figure 25). The spatial layer provides distances to the nearest trapping site. Traps are assumed to be effective to 3.5 kilometers. Cowbirds with breeding territories within the range of traps are designated 'at risk' and experience lower survival rates than individuals out of range of traps. Figure 27 includes maps of modeled vireo and cowbird habitats and cowbird trap locations.

Several other components of the vireo-cowbird population model merit further description. First, the hexagon grid is based on 0.56-ha hexagons. At this resolution, vireos in the highest quality habitat will require two hexagons to form a territory, approximating the highest vireo density recorded during point counts on Fort Hood (0.68 individuals / ha). Cowbird breeding territories are larger than vireo territories and therefore less affected by grid cell size. Second, the modeled vireo population is males-only, whereas the modeled cowbird population is females-only. All density data for Fort Hood refers to male vireos and productivity estimates are per male territory (Goering 1998). Therefore, reproduction is modeled per male. Estimates of cowbird fecundity are per female (Payne 1976, Scott and Ankney 1983). Third, female cowbirds will be modeled in breeding territories and not on their foraging grounds. Woodland habitats within 600 m of grasslands are treated as cowbird breeding habitat. Parasitism of vireo nests has been shown to correlate with nest density of other hosts (Barber and Martin 1997). However, for simplicity, cowbird reproduction will remain constant.

Model Calibration. The vireo-cowbird population model is complex and includes 38 parameters (Table 8). Parameter values were based on published estimates where available, unpublished monitoring data from The Nature Conservancy of Texas' Fort Hood Project (TNC-FHP), and expert opinion. Parameterization was completed iteratively by comparing model output to vireo population abundance estimates (Cimprich 2009). Each model run included a 4-timestep spin-up in which the initialized vireo population adjusts to the distribution and parasitism pressure from the cowbirds. The simulated distribution and abundance of vireos at time-step 4 is an estimate of the *ca.* 1990 pre-cowbird trapping population. Time-steps 5-19 simulate the 15 years of base-wide cowbird control including both cowbird trapping and shooting. Time-steps 20-25 simulate the cessation of cowbird control on the west range of Fort Hood. Model fit was assessed by the model's ability to capture 1) the rise in vireo population to near carrying capacity by the end of the period of base-wide cowbird trapping, and 2) the decline to a new population level following the cessation of cowbird trapping on the west range. No data were available for calibration of

the cowbird population model. Model fit was assessed indirectly through its effect on the vireo population.

Sensitivity Analysis. The complexity and slow processing speed of the VCPM encumbers attempts at a global sensitivity analysis using factorial or grid-search methods requiring thousands of model runs. Therefore, a variable screening methodology, the Morris method, was used to rank the influence of input parameters on model output (Saltelli 2004). The Morris method randomly generates a series of ‘walks’ through parameter space in which one variable is changed for each step and each step requires a model run. Model output from the walks are used to calculate an elementary effect for each variable based on the size of the step (in standardized units) and the impact on the model output. Elementary effects for the same variable but from multiple walks are summarized to characterize the impact of each parameter on model output. The Morris method has proven nearly as effective as more computationally expensive options at ranking parameters by sensitivity (Confalonieri et al. 2010).

The Morris Method was completed on an early version of the VCPM. Seventeen parameters were varied and ten ‘walks’ conducted, requiring 180 VCPM runs. Comparative analyses have identified ten as an optimum number for parameter ‘walks’ for characterizing sensitivities (Saltelli 2004). The mean of the absolute values of each parameter’s elementary effects assesses a parameter’s overall influence on model output (μ^*), while the standard deviation measures of the degree to which a parameter’s influence is impacted by interactions with other parameters (σ).

Model Scenarios. Two cowbird-trapping scenarios were run in combination with output from the Fort Hood Vegetation Growth and Disturbance Model (Table 9) for a total of ten VCPM scenarios. The cowbird trapping scenarios include base-wide trapping and shooting and no trapping or shooting. In the base-wide trapping and shooting scenario, traps are distributed throughout grassland areas across Fort Hood and three shooting routes are patrolled (Figure 27). The VCPM was run for 15 vegetation projections representing each scenario. Five replicate runs were completed for each of these vegetation projections. Across all scenarios, a total of 750 runs of the VCPM were completed.

Table 9. Scenarios for the Vireo-Cowbird Population Model. The scenario including base-wide cowbird control was run with all output scenarios from the Fort Hood Vegetation Growth and Disturbance Model, whereas the no-control scenario was run with a subset.

Scenario	Military Training	Fire	Cowbird Control
Baseline	None	Historic probabilities	Base-wide, None
Train West Range	West Range	Historic probabilities	Base-wide, None
Open East Range	West & East Range	Historic probabilities	Base-wide
Larger Fires	West Range	Increased fire extents	Base-wide, None
Fire Suppression	West Range	Historic probabilities, but no fires outside of LF	Base-wide
Train West & East Ranges and Larger Fires	West Range	Increased fire extents	Base-wide, None

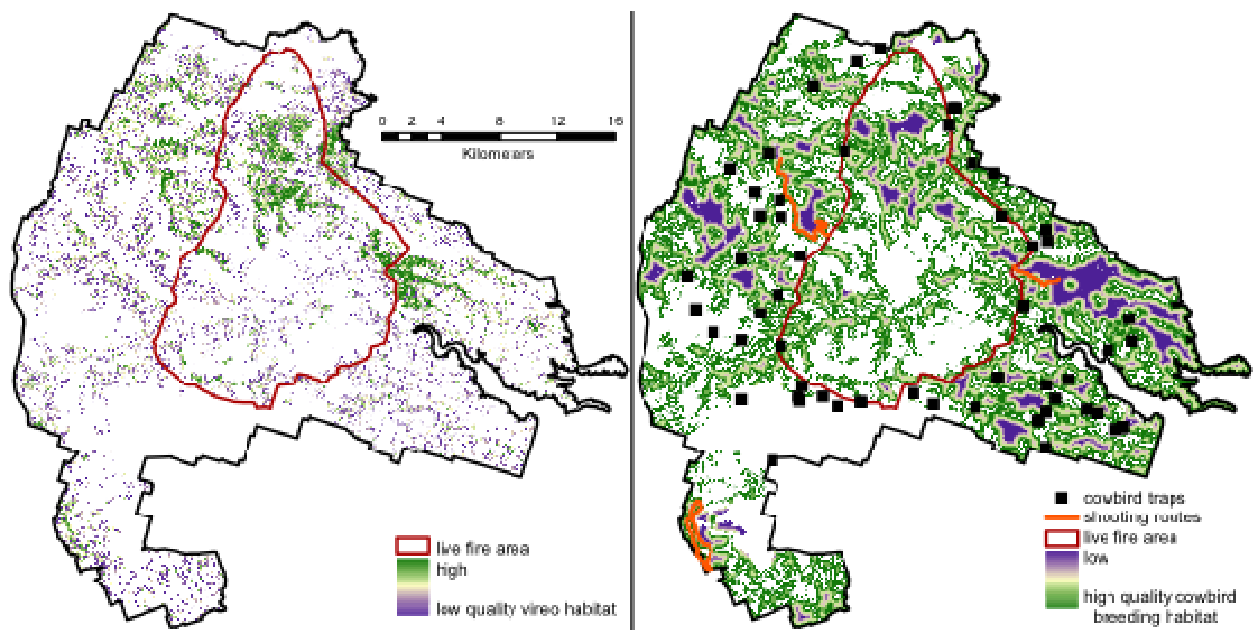


Figure 27. Map of probability of vireo occurrence produced by the Vireo Habitat Suitability Model (left panel). Grid cells with probabilities below 0.369 are considered unsuitable. Cowbird trapping sites, shooting routes, and breeding habitat modeled as proportional to distance from grasslands (right panel).

The Desert Tortoise Case Study

Desert Tortoise Habitat Model

Unlike the potential habitat relationships for the RCW (see the section titled *Red-cockaded Woodpecker Case Study*), no evidence suggests that desert tortoises select or use habitat at multiple spatial scales. In addition, desert tortoises are extremely difficult to locate under field conditions. Presence locations are reliable, but previous research suggests surveys underestimate the number of tortoises within an area (Anderson et al. 2001). Therefore, absences are difficult to generate or verify for use in typical habitat modeling methods.

At the beginning of the current study, no detailed desert tortoise habitat model was available. Andersen et al. (2000) built a model predicting desert tortoise habitat on the southern side of Fort Irwin. However, this model was not readily available and did not cover the entire extent of the current study. A reliable model was necessary to project the effects of URTD and precipitation on desert tortoises in the Western Recovery Unit. MaxEnt, short of Maximum Entropy, was used to model habitat for desert tortoises in the Western Mojave Recovery Unit.

Desert tortoise presence locations were obtained from deserttortoise.gov on 13 January 2009 and the line distance sampling data from 2001-2005 were used. In addition, records of desert tortoise observations from Fort Irwin were included, for a total of 2,728 presence records. Soil characteristics were important predictors of desert tortoise habitat in a previous model (Andersen et al. 2000). Therefore, the soil texture class from CONUS soil data (Miller and

White 1998) was included in the model. Separate data layers of the dominant soil texture at 5, 10, 20, 40, 60, 80 and 100 cm were generated from the CONUS data within ArcMap 9.3 and clipped to the study extent. Slope and aspect layers were generated within ArcMap 9.3 from a 1/9 arc-second Digital Elevation Model (DEM) obtained from <http://seamless.usgs.gov/ned19.php> and clipped to the study extent. Vegetation cover data were obtained from Thomas D. Frank (Tweddle and Frank 2006).

MaxEnt is a machine learning approach which uses presence-only data to model species distributions (Phillips et al. 2006). MaxEnt has been shown to perform well on presence-only data and allows the modeling of non-linear relationships between species presence and habitat characteristics. Thirty percent of presence records were randomly withheld as test data, while 10,000 background points were generated within MaxEnt to characterize areas without known presences. Area under the ROC curve (AUC) was used to assess habitat model performance. Predictor variable importance was assessed within MaxEnt using a jackknife technique that compared output from paired habitat models in which each predictor variable was excluded and included as the only predictor.

Simulating Population Responses to Upper Respiratory Tract Disease and Climate Change

HexSim was used to simulate the potential interaction between URTD and drought conditions on the desert tortoise population. The desert tortoise population was simulated from the year 2000 to the year 2100 under three scenarios: 1) without either stressor, 2) with each stressor alone, and 3) with both stressors in combination.

Potential impacts of climate change were based on simulated precipitation data. Precipitation data for the years 2000-2100 were projected for three emissions scenarios using three different GCMs (see the section titled *Climate-change projections*). Total annual precipitation for September through March from each climate model was summed per pixel of the study extent. Maps of total fall-spring precipitation for each year were input to HexSim as a spatial layer. The average amount of September-March rainfall was calculated for each tortoise within its home range each year.

Parameterization of the HexSim tortoise model relied heavily on the parameters used by Doak et al. (1994), but were modified slightly for HexSim. Each simulation was allowed to stabilize (the “burn-in period”) for 30 years prior to introducing new landscapes or other data into the model. Baseline population trajectories were targeted to match approximate tortoise density in the Western Mojave (U.S. Fish and Wildlife 2008). However, to model tortoise densities in the absence of either drought stress or URTD, the population was parameterized to have a $\lambda > 1$ (adult female survival = 94%).

To model the effects of drought alone, survival was reduced in adult and sub-adult classes by 2% in years dropping below the 25 mm total precipitation (failure of winter annuals). This may be a conservative estimate of the effect of drought on adult survival, or may overestimate the effects of drought alone. Previous research suggests the range may be between 0% mortality to 10% mortality (Longshore et al. 2003). The baseline parameter estimate for adult survival was 94%, so reducing adult survival by 2% kept estimates of survival within the 90-100% range when drought was present.

To model the effects of disease alone, annual survival of adult and sub-adult classes was set between 91% - 93.75%. The model was not parameterized to model URTD infection between tortoises, as transmission coefficients for URTD in desert tortoises are unknown. Thus, the model was parameterized using the population estimates for survival rates (the average

survival of any tortoise, assuming URTD is present in the population). These estimates were extracted from published literature (Turner et al. 1984, Longshore et al. 2003).

To model the potential synergism between drought and URTD mortality, an empirical linear relationship between survival when URTD was present in the population and September-March precipitation in the Western Mojave was used to create estimates of survival rates for any particular average precipitation level (Figure 28; Peterson 1994, Berry 1997, Christopher et al. 1997, Mullen and Ross 1997).

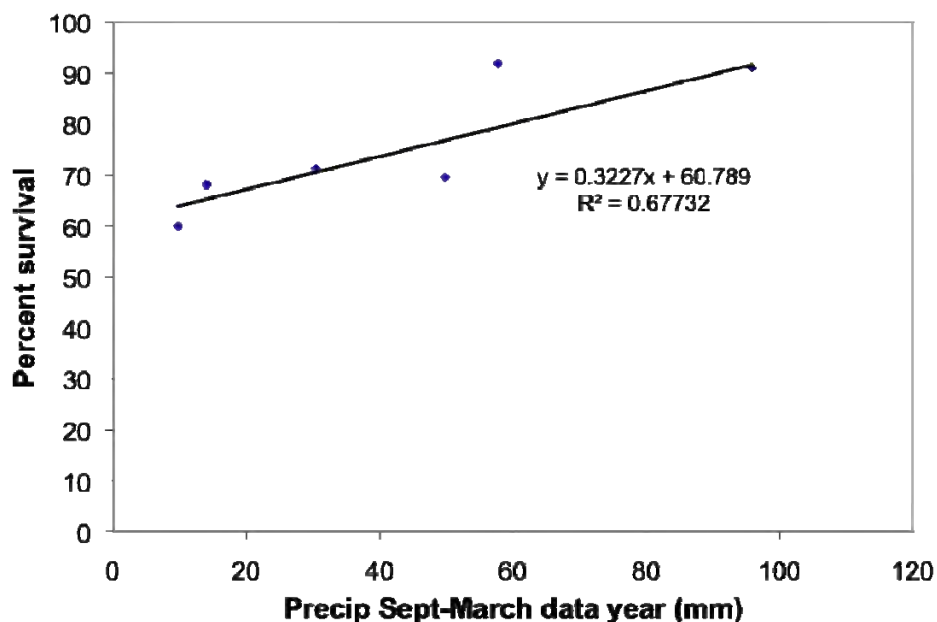


Figure 28. Empirical relationship between September-March rainfall and survival rates in populations with URTD present in the Western Mojave (data from Peterson 1994, Berry 1997, Christopher et al. 1997, Mullen and Ross 1997).

Objective 3: Provide the DoD with the improved, highly flexible, user-friendly modeling tool, a comprehensive user's manual, and a training workshop on using the model.

HexSim and the user's manual were developed and made freely available on-line to the DoD and the general public. The model was presented to DoD's Conservation Committee on May 11, 2010.

Results and Discussion

Objective 1: Extend an existing spatially explicit population model, enabling it to assess the effects of multiple interacting stressors on population dynamics.

Interactive Stressors Module, Interspecific Interactions Module, and Improved Model Outputs.

The description of the results of these research activities can be found in the **Materials and Methods** section. The choice was made to report these results as methods because they provide the background for the methods used in the three case studies. It was determined that the reader of the report would need a basic understanding of the HexSim model to best understand the methods used in the three case studies.

Additional Model Features

The goal of this effort has always been a flexible, powerful, but user-friendly model. Merging these divergent design goals meant providing access to a great array of sophisticated algorithms, but not insisting they be applied. It meant allowing users to determine what level of detail to represent in a simulation, and hence how much data would be required. Finally, it meant supplying a modern graphical user interface for every aspect of the model. And this interface had to be consistent, so that insights gained using one module could be applied later to another.

It is not possible to construct a useful ecological model in a vacuum, especially one as complex as HexSim. For that reason, graduate students, postdoctoral fellows, and researchers from a variety of institutions were recruited to test the evolving model. Foremost among this group were the members of the Lawler laboratory at the University of Washington, who were conducting the three case studies that formed the bulk of this SERDP-funded research effort. This consortium worked for years, testing existing algorithms, identifying un-met needs, developing new HexSim modules, enhancing the model interface, fixing coding errors, and so on. The result of these efforts are a final product that has many more features than were originally anticipated. The current version of HexSim has also been subjected to years of intense scrutiny to insure its accuracy and identify and fix coding errors. No complex model is free from bugs, but HexSim has been subjected to vastly more quality assurance testing than most other ecological models. Some of the model features not anticipated in the original SERDP proposal are discussed briefly below.

The Model Interface

In overly-simplistic terms, HexSim consists of a graphical user interface (GUI) and a model engine. SERDP funding supported the development of the model engine, the construction of which extended for the life of project RC-1541 (and beyond). Throughout this entire period, the EPA matched this funding by providing a contractor who developed the model GUI. The GUI is what users interact with—users do not work directly with the model engine.

There are many features of the HexSim interface that go beyond what was originally envisioned when the RC-1541 project proposal was drafted. These are items provided for user convenience, such as a history of recently opened workspaces, tools for adding and copying workspaces and scenarios, ways to recover data from memory or disk files, etc. Users can now color events, copy events, and reorder trait combinations. The model interface performs error checking, and flags events that are incomplete or incorrectly parameterized. This represents just a few examples of an effort that spanned multiple years, and was dedicated to building tools usable by scientists and non-scientists, students and professionals, researchers and managers, etc.

Multiple Color Schemes

A frequent complaint of PATCH, and earlier versions of HexSim, focused on the gradient of colors that had been selected to represent habitat quality, or model outputs such as occupancy rate. As it turned out, the original color scheme was likely the worst choice possible for individuals who suffer from red-green color blindness. To remedy this problem, users are now allowed to select from four different color gradients. This seemingly small change (it was not trivial to implement) has improved the model substantially for a subset of our user-base.

Event Triggers

Event triggers are another example of a minor enhancement that added substantially to the model. HexSim's triggers allow individual events to fire once, during a specific window of time steps, or periodically within starting and ending points. Triggers make it possible for new individuals to be introduced to a population at predetermined times. They allow perturbations to take place on a schedule, as opposed to every time step. For example, triggers can be used to allow a population to reach steady state before the introduction of a disease or parasite. Triggers also make it possible to perform special tasks when a simulation starts up, but not afterwards.

The Workspace Constructor and HexMap Generator

The PATCH model, which HexSim grew out of, had a single tool that built new workspaces and new HexMaps. HexSim breaks these into two distinct functions, and in doing so makes both simultaneously more powerful and more usable. Splitting these tools necessitated re-writing both modules entirely. The HexSim workspace constructor is now intuitive and extremely fast to use. The HexMap generator no longer has to be told what grid parameter to use (these are established by the Workspace Constructor), and it is singly-focused on the task of HexMap creation. At the same time, the new tool is more flexible and more robust than what was available previously. The HexMap Generator can now also construct blank HexMaps that can be modified using HexSim's new HexMap editing tools, and used in more theoretical analyses.

HexMap Editing

HexSim now includes a tool for directly editing HexMaps. Users may select a new hexagon value, and then, using the mouse, apply this value to individual hexagons, hexagons along a line, within a rectangle, or within a polygon. This editing tool makes certain kinds of investigations much easier to conduct than they were previously. For example, hexagon editing can be used to develop maps of stressor distribution, or to add or subtract habitat to simulate resource extraction or restoration activities.

HexMap Smoothing

HexSim's smoothing algorithm runs a moving window through a HexMap and assigns the mean or sum taken across the window to the focal hexagon. Users select the window size and a threshold value below which hexagons are left out of the analysis. HexMap smoothing is mainly useful for generating HexMaps that can be used with a Movement event to guide individual dispersers towards high value portions of a landscape. HexMap smoothing is available both from the Workspace Spatial Data context menu, and from within the Generated HexMap event.

HexSim Barriers

Movement barriers have significant impacts on wildlife populations in the real world. In spite of this well-appreciated fact, they are often not represented in ecological models because doing so is very difficult. In an effort to add this important realism, a great deal of time and effort was invested in developing a robust barrier toolkit for HexSim. HexSim's barriers are extremely flexible. They can be imported from an ESRI shapefile, or added directly through the model interface. Barriers can be static, or packaged as time series. Multiple barrier series can be used within a single simulation, and multiple barrier types can be present within any single barrier series. Each barrier type has unique parameter values that specify the probabilities of mortality, deflection, and transmission. Further, HexSim barriers are two-way data structures, so separate mortality, deflection, and transmission properties can be associated with crossing from different directions.

HexSim barriers can affect movement and range formation. But these impacts take place on an event-by-event basis. So, for example, it is straightforward to develop one set of barriers that will impact an early life stage, but also have an entirely different set used to influence older members of the same population. One-way barriers might be used to represent a cost to entry into an urban area, yet impart no penalty for exiting. Purely transmissive barriers can be used to track dispersal fluxes without interfering with a population. These are just a few examples of this rich but likely under-appreciated component of the HexSim model.

Importing and Exporting

Three import functions have been added to the HexSim interface. These are for importing scenarios from other workspace, and for importing HexMap and Barrier map data. HexMap data can be imported from CSV (comma-separated variable), TXT (text), or DBF (typically part of an ESRI shapefile) files. Barrier maps can be imported from SHP (part of an ESRI shapefile), or HBF (HexSim barrier format) files. HexSim spatial data can be exported to bitmap files, and ESRI shapefiles. HexSim now also provides a translation tool that lets users shift back and forth between HexSim and ESRI coordinate systems.

Batch Files and BatchRunner.exe

It is common for users to want to set up a large number of simulations, and then run them as a batch process. HexSim now contains a batch processing module that makes this possible. Tools exist to create and edit a batch file. Once the batch file is complete, users launch a separate executable called BatchRunner, and use it to run the desired simulations. BatchRunner can execute simulations both in series and in parallel.

Sensitivity Analysis

HexSim now includes a sensitivity analysis toolkit as a stand-alone executable that allows users to pick a parameter and then specify a range of values that it should iterate through. This can be done for each numeric model parameter that is not in some way constrained by the value of another parameter. The sensitivity analysis tool develops a dedicated batch file, which can later be executed using BatchRunner (see above).

HexSim Genetics

A full genetics sub-model has been added to the final HexSim product. HexSim's genetics are an extension of its trait superstructure. The genetics tool allows populations to be

assigned a genome consisting of multiple loci. Any number of alleles can be associated with individual loci, and there are multiple ways to allocate alleles at model initialization. Two mating schemes have been added to HexSim, and both can be used to develop offspring genotypes from the parent's genetic data. HexSim genetics includes a tool for simulating crossover based on map distances, and also a Mutation event. HexSim genetics can be used to develop adaptive traits that influence vital rates and behaviors.

Special HexSim Events

A number of specialized events have been developed to extend the range of problems that HexSim may be applied to. Examples include the Accumulate event, which includes a wide range of Updater Functions. Some of the more specialized updater functions allow users to transfer accumulator value from one individual to another, to track the reproductive output for individual females, track group size, locate individuals within a landscape, break up pairs in which one member has died, quantify local environmental quality, increment updaters stochastically, and so on. Other examples of specialized events include Introduction and Set Group Affinity. The Introduction event allows individuals to be added to a population after a simulation has already started. The Set Group Affinity event can be used to move flocks, packs, or herds, etc. as a group.

HexSim Affinities

Affinities are constructs that allow individuals learn from their past experiences. HexSim has four types of affinities: Natal, Reproduction, Resource, and Group Movement. Group movement affinities provide a mechanism for moving an entire group of individuals towards a common dispersal target. The Set Group Affinity event is used to identify an affinity site and assign it to all members of a group. If those group members subsequently become floaters and move, this group movement affinity can then be used to direct each individual towards the same target site.

Natal affinities record the birth hexagon. If the individual is born into a multi-hexagon range, the most central hexagon in the range is used as the birth hexagon. Natal affinities cannot be altered. Reproduction affinities record sites at which reproductive output was high. This affinity data is associated with the parent, but the location is the same as the offspring's natal hexagon. The site that is selected for the affinity record will be the most central hexagon in the individual's range. Resource affinities record sites with high resource availability. Resource affinities are stored by the exploration component of Movement events. If the exploration goal includes starting or joining a group, then only individuals who successfully become group members will store resource affinities. The site that is selected for the affinity data will be the most central hexagon in the individual's new range.

Global Assignment and Show Usage

Sometimes many changes must be made to a scenario all at once. For example, it might become necessary to substitute one spatial data time series for another throughout a scenario. If many events use this spatial data, then doing so one parameterization window at a time can be tedious and time consuming. The Global Assignment tools each allow users to change one type of data throughout a simulation, using a single dialog window. Global Assignment tools are available for spatial data (HexMap time series), barrier time series, affinities, log parameters, and descriptions.

To provide users with additional help managing complex scenarios, a Usage Summary tool has also been included with HexSim. This tool provides users with a summary of all of the important dependencies they have built into a scenario. Users can see every use of a particular spatial data time series, or barrier series. They can see which traits and trait values have been used to parameterize populations and events, and what uses have been made of a population's accumulators. Event's use of affinities and genetic traits can be displayed as well.

Objective 2: Evaluate the relative and cumulative impacts of climate change, environmental stochasticity, military activities, and other species- and site-specific threats on three at-risk wildlife populations, at three DoD sites.

Climate-Change Projections

Fort Benning

Projected increases in mean annual temperature for the Fort Benning study area from the 30-year mean period of 1961-1990 to the 30-year mean period of 2070-2099 vary from a low of 1.58 °C under the B1 emissions scenario to a high of 4.66 °C under the A2 emissions scenario (Shafer et al. In-Press). Projected changes in mean annual precipitation for the same period vary from a low of a 1% increase to a high of a 23% increase.

Fort Hood

Projected increases in mean annual temperature for the Fort Hood study area from the 30-year mean period of 1961-1990 to the 30-year mean period of 2070-2099 vary from a low of 1.86 °C for the B1 emissions scenario to a high of 5.21 °C for the A2 emissions scenario (Shafer et al. In-Press). Projected changes in mean annual precipitation for the same period vary from a 5% decrease to an 18% increase.

Fort Irwin

Projected increases in mean annual temperature for the Fort Irwin study area from the 30-year mean period of 1961-1990 to the 30-year mean period of 2070-2099 vary from a low of 2.19 °C for the B1 emissions scenario to a high of 4.74 °C for the A2 emissions scenario (Shafer et al. In-Press). Projected changes in mean annual precipitation for the same period vary from a decrease of 21% to an increase of 56%. Seasonally, the study area is projected to experience decreases in winter (0-25%) and spring (11-38%) precipitation and increases in summer precipitation—from a decrease of 17% to an increase of 477%. The large percent increases projected in some of the climate-change scenarios reflect the fact that this region experiences relatively little summer precipitation and so small increases in precipitation can translate into large percentages.

Vegetation-Change Projections

Fort Benning

The LPJ model simulated a mixed evergreen and broadleaf forest for the Fort Benning study area for the 30-year mean period from 2070-2099 (Shafer et al. In-Press). This

corresponds relatively well to the oak-hickory-pine forest described as a likely dominant natural potential vegetation type for the region (Küchler 1993). Similarly, LPJ simulations resulted in projections for mixed evergreen and broadleaf forests for the end of the 21st century under all nine future climate projections. Other vegetation modeling studies have produced similar results for the same region (Bachelet et al. 2008, Lenihan et al. 2008).

The LPJ-GUESS model simulations also projected little change in vegetation across the Fort Benning study area. Under all 9 climate scenarios, the model projected needle-leaved evergreen basal area of $\sim 35 \text{ m}^2 \text{ ha}^{-1}$ for the 30-year mean period from 2070 to 2099 for trees >20 years old. This was similar to the simulated basal area for the 30-year mean period from 1961 to 1990. Simulated needle-leaved evergreen tree density ranged from 450 trees ha^{-1} for the B1 emissions scenario to 453 trees ha^{-1} for the A2 emissions scenario for the 30-year mean period from 2070 to 2099 for trees >20 years old. Again, these densities were roughly the same as those simulated for the period from 1961-1990.

Despite the projected increases in temperature for the study area, it appears that Fort Benning and its surroundings will still be suitable for longleaf pine by the end of the 21st century. The largest projected change in temperature explored in these simulations would result in an average annual temperature of 22.46 °C, which is still within the range of historical mean annual temperatures experienced by longleaf pine in the southeastern United States. Increases in fire frequency that will likely result from higher temperatures and increased evaporation have the potential to benefit the fire dependent long-leaf pine ecosystem and in turn RCW habitat.

Although these model projections forecast the persistence of evergreens and likely long-leaf pine in particular, these models were not calibrated to project changes in population of individual species. Nor do these models account for changes in pests and pathogens or the introduction of dominant competitors that may occur over the coming century.

Fort Hood

Vegetation for the Fort Hood study area was simulated both with and without fire (Shafer et al. In-Press). When fire was included in these simulations, both the LPJ and the LPJ-GUESS models simulated grasslands and savannas for the study for the historical (1961-1990) period. These projections roughly match description of historical vegetation in the region—grasslands and savannas with denser tree cover found in more mesic areas, such as along riparian corridors (Reemts and Hansen 2008). When applied to future climate projections LPJ simulations resulted in mixes of

needle-leaved evergreen, broadleaved summer-green and evergreen, and grasses. Grass cover increased for five of the nine climate-change scenario combinations and woody plant cover increased for the other four climate-change scenarios. LPJ-GUESS simulations produce the same mix of PFTs under projected future climates. However, LGP-GUESS simulations were quite sensitive to fire. When fires were simulated, the grass cover increased to $\sim 65\%$ and woody plant cover was reduced to $<5\%$ under all nine climate-change scenarios.

A second set of LPJ-GUESS vegetation simulations that excluded fire were conducted to simulate potential impacts of climate on woody vegetation. These runs provided information on the response of woody vegetation to future climate changes and atmospheric CO₂ concentrations in the absence of fire. In these simulations, grass cover decreases to $\sim 10\%$ and woody plant cover increases to $\sim 46\%$ under all nine climate-change scenarios. For broadleaved woody plants <20 years old, the suppression of fires in the model doubled the percent cover under all nine climate-change scenarios. These simulations emphasize the importance of fire to the vegetation

in and around Fort Hood. Given that fire frequencies and intensities are likely to increase in the western US, this region may see changes in the relative amount of Black-capped Vireo and Golden-cheeked Warbler habitat. Increases in the frequency of fire could be beneficial to the vireo, keeping woody vegetation short and suitable for nesting. Such a change would reduce the number of taller trees and hence warbler habitat. However if fires are too intense and too frequent, grasses could come to dominate the area, reducing habitat for both species.

Fort Irwin

For the historical time period, LPJ simulations resulted in a mix of four PFTs across the Fort Irwin study area (Shafer et al. In-Press). These included needle-leaved evergreen, broadleaf evergreen, broadleaf deciduous trees/shrubs, and C3 grass. Again this vegetation is similar to that suggested by the potential natural vegetation types defined by Küchler (1993). Although the model simulated the occurrence of multiple PFTs, it simulated relatively little cover overall. LPJ simulated 23% vegetative cover for the study area for the historic period. The remaining area was simulated as bare ground. Observed percent cover for perennial plants at Fort Irwin has been reported to range from 4% to 28% (Berry et al. 2006).

LPJ model simulations for the averaged 30-year period from 2070-2099 indicated that the study area would likely experience a similar mix of vegetation types. However, the model simulated an increase in the percent cover of grasses. The increase in grass cover may be driven by increases in water-use efficiency resulting from increased CO₂ concentrations or from increases in summer precipitation. It is important to note, however, that because the model simulates PFTs and not individual plant species, the projected increase in grass cover could manifest itself as native or non-native grasses.

The projected increases in temperatures and decreases in winter precipitation have the potential to impact the desert tortoise both directly and indirectly through changes in the fire regime. Increases in fire frequency potentially pose a threat to tortoises. Fires are projected to become more frequent for all nine climate-change scenarios. Under three scenarios, the fire return interval for 2070-2099 is projected to be 13 years—roughly half of the current fire return interval. Increased future fire occurrence could lead to increased desert tortoise mortality. Recent observations indicate that increases in fire frequency have likely already begun as a result of the invasion of non-native and invasive plant species.

Because these climate-change projections are for changes in monthly averages, they ignore daily variability in precipitation and temperature. Furthermore, they don't provide projected changes in the frequency and intensity of precipitation events, which may affect how much drinking water is available to desert tortoises and how much vegetation is available for forage and the quality of that forage. In addition, GCMs generally are less adept at modeling precipitation than they are at modeling temperature. Given the important role that precipitation plays in the life of the tortoise, our ability to project future climate impacts is in part restricted by the ability of the GCMs to simulate precipitation changes.

Red-cockaded Woodpecker Case Study

Multi-Scale Habitat Model

Although the models based on the three different approaches (Random Forest predictors, logistic regression, and generalized additive models) identified some common regions with high probability of being RCW habitat, they differed in their predictions in some areas (Figure 29).

The total percentage of the base predicted to be habitat varied among the models, with the largest area predicted by the ensemble of three different modeling approaches applied at the foraging scale (49.6 % of area predicted suitable), and the smallest area predicted by the generalized additive multi-scale model (26.6% of area predicted suitable habitat). In general, the models performed relatively well, with AUC ranges between 0.92-0.98 (Fort Benning; Table 10) and 0.83-0.91 (Fort Bragg; Table 11).

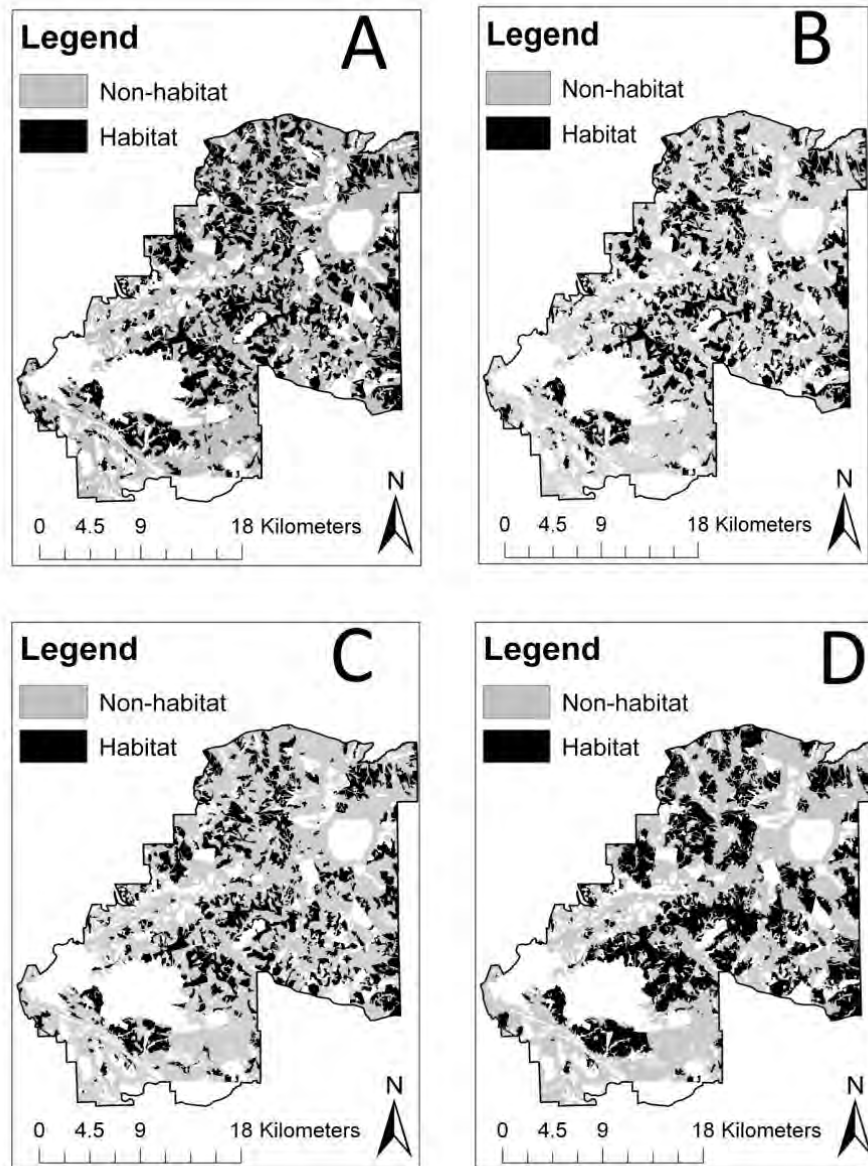


Figure 29. Comparison of predicted presence models from three modeling approaches and the ensemble of the three approaches. Panel A is the predicted presence from a Random Forest model, B is from a logistic regression model, and C is from a generalized linear model. Panel D is the model approach ensemble at the nest scale (weighted average probability of presence). White areas indicate missing or incomplete data.

Table 10. Model performance from test data at Fort Benning, GA. Thresholds were selected using the MaxKappa function in the R package PresenceAbsence. Data for ensembles are calculated from weighted averages, followed by values from unweighted averages in parentheses if different from weighted averages. RF = Random Forest, LR = logistic regression, GAM = generalized additive model.

Class	Scale	Modeling approach	AUC	Percent test data correct	
				Presence	Absence
Single models	Nest	RF	0.93	93	83
		LR	0.90	85	87
		GAM	0.92	87	86
	Forage	RF	0.96	92	85
		LR	0.92	91	79
		GAM	0.92	81	86
	Multi-scale	RF	0.97	95	89
		LR	0.94	83	92
		GAM	0.93	85	93
Model approach ensembles	Nest	All	0.93	96	78
	Forage	All	0.94	94	78
	Multi-scale	All	0.96	87	93
Scale ensembles	Nest and Forage	RF	0.98	96 (92)	92 (94)
		LR	0.94	93	85
		GAM	0.95	91 (95)	86 (82)

Table 11. Model performance from test data at Fort Bragg, NC, USA. Thresholds were selected using the MaxKappa function in the R package PresenceAbsence. Data for ensembles are calculated from weighted average probability, followed by values from unweighted averages in parentheses if different from weighted averages. RF = Random Forest, LR = logistic regression, GAM = generalized additive model.

Class	Scale	Modeling approach	AUC	Percent test data correct	
				Presence	Absence
Single models	Nest	RF	0.83	61	97
		LR	0.84	61	97
		GAM	0.86	90	66
	Forage	RF	0.91	78	86
		LR	0.85	77	73
		GAM	0.87	73	86
	Multi-scale	RF *	0.91	80	87
		LR	0.83	92	57
		GAM	0.83	87	61
Model approach ensembles	Nest	All	0.84	62	97

Scale ensembles	Forage	All	0.88	77	84
	Multi-scale	All	0.86	73	81 (80)
	Nest and Forage	RF	0.90	71 (72)	95
		LR	0.87	82	76
		GAM *	0.91	79 (83)	90 (85)

* Indicates at least 80% accuracy in both presences and absences calculated using either weighted or unweighted average probability.

Foraging-habitat models tended to be better predictors of occurrence than did nesting-habitat models (Tables 13 and 14). Although multi-scale habitat models tended to be more accurate than either of the single-scale models when tested on the Fort Benning data, they were not necessarily more accurate than the single-scale models when applied to the Fort Bragg data (Table 11).

Random Forest models were more accurate than logistic regression or generalized additive models across scales when applied to the Fort Benning data (Table 10). Random Forest models applied to Fort Bragg data were more accurate than the other two approaches for foraging-habitat and multi-scale models, but not for nesting-habitat models (Table 11). Patterns in omission and commission error also differed among modeling approaches (Tables 13 and 14). For example, all Random Forest models had higher rates of omission (failure to predict locations of observed presences) than commission (prediction of false presences) when applied to the fully independent Fort Bragg data. With the exception of the nesting-habitat model, all logistic regression models had higher rates of commission than omission when applied to the Fort Bragg data.

Scale ensembles had higher AUC values than the single-scale nesting-habitat models and foraging-habitat models when applied to the Fort Benning data (Table 10). These ensemble models also had marginally better or equal AUC values when compared to the multi-scale single models (Table 10). In general, scale ensembles also outperformed the individual scale models when applied to the Fort Bragg data. AUC values and accuracy were higher for scale ensemble models compared to single models for logistic regression and generalized additive models, including the multi-scale single models (Table 11). In contrast, model performance of the Random Forest multi-scale single model was better than the Random Forest scale ensemble at Fort Bragg (Table 11).

The ensembles of different modeling approaches did not consistently outperform the individual models. In general, Random Forest models had higher AUC values than the ensembles of the various modeling approaches with similar rates of omission and commission errors (Tables 13 and 14).

All models, regardless of scale or approach, performed well on both sets of test data. Good performance on independent test data from Fort Bragg, North Carolina, suggests that the models are transferrable in space, and did not overfit the data from Fort Benning. Generally, models built using Random Forest and the scale ensembles performed better on independent test data.

No improvement in model performance was observed by combining the predictions of different modeling approaches. This is likely due to the high performance of the models built using Random Forest. The modeling approach ensembles performed better than generalized additive models and logistic regression models, but worse than the Random Forest models. It is possible that the Random Forest approach was more accurate because it successfully modeled important habitat factors for RCWs. However, the two other algorithms also performed well, suggesting that all of the models likely captured some of the important factors. It is also possible

that the Random Forest models performed as well or better than the ensembles of different model approaches because Random Forest is itself an ensemble technique. Random Forest uses a slightly different approach to create ensemble models than we used here. Instead of combining multiple scales or multiple approaches, Random Forest uses a subset of the predictor variables and observations to build each individual tree model. This approach results in models built using different starting conditions, or permutations, for each tree in the forest. Random Forest has previously been shown to be a promising technique for modeling species distributions (Cutler et al. 2007) and appears to work well for finer scale habitat modeling.

Ensemble models built from nesting- and foraging-habitat models (scale ensembles) had improved model performance compared to multi-scale models (single models that incorporated predictor variables from both the nesting and habitat scales). Many organisms select and use habitat at multiple scales, and our work suggests that incorporating the relevant scales into habitat models can be efficiently accomplished through the use of ensemble techniques. It is possible that the ensemble models performed better because modeling each relevant scale alone better captured the processes occurring at each scale. For instance, in the generalized additive nest model, age (60), basal area of pine, and total basal area were included. In the generalized additive forage model, age (30), age (60), conifer, and percent pine were included. However, only age 30(forage), age 60(nest), conifer (forage), total BA (nest) and percent pine (nest) were included in the multi-scale single model. Thus, using an ensemble technique to model habitat may be a more efficient way to model processes occurring at different scales. In contrast, model performance of the Random Forest scale ensemble was not better than the Random Forest multi-scale single model, suggesting that some multi-scale models may perform as well as ensembles in some cases. Again, however, this may be explained by the fact that the Random Forest models are, themselves, an ensemble approach. The results suggest that generating ensembles of models built with data collected at different spatial scales should be considered as a useful approach for modeling multi-scaled processes.

Generally, all models in this study performed well. As all models performed well, but Random Forest was often the “best” model and is an ensemble technique, all maps for all RCW simulations were generated using Random Forest.

Population Simulations in HexSim

Simulating Loblolly Decline. All management scenarios resulted in eventual RCW numbers at or above the current total of 275-300 potential breeding pairs (Figure 30). However, the interplant management scenario resulted in the most potential breeding pairs (PBP) and exceeded the Fort Benning recovery goal of 351 potential breeding pairs on the landscape (average PBP \pm SE: 367 \pm 10).). The cut management scenario resulted in many fewer breeding pairs during years 2010-2085, but then gained quickly to exceed all scenarios except for the interplant scenario. The no restoration scenario resulted in less than half the number of PBP than the recovery goal (average PBP: 135 \pm 8).

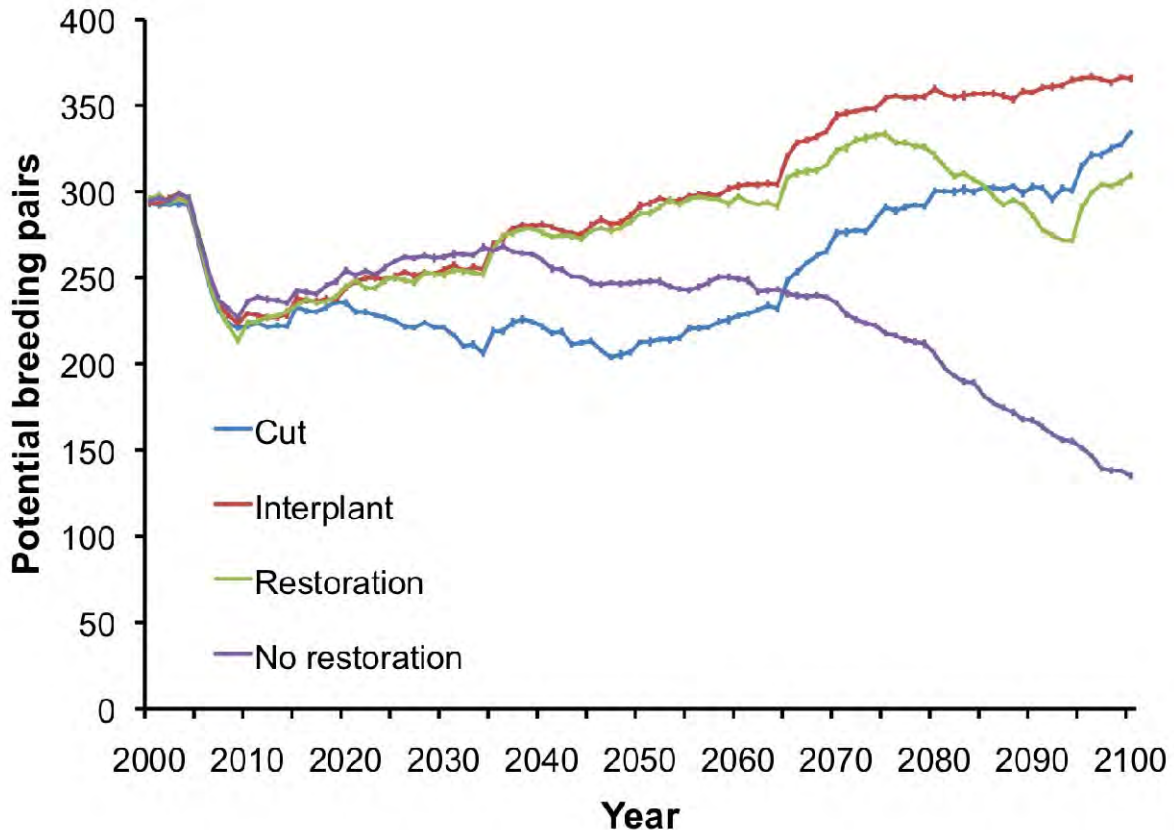


Figure 30. Comparison of loblolly decline management scenarios for Red-cockaded woodpeckers (RCW) at Fort Benning.

Given the RCW's dependence on habitat structure, it was not surprising that the largest negative effects of any future scenarios were observed under those scenarios in which habitat was extensively altered. These negative effects were observed both due to land-use change, particularly the Worst-case scenarios, and due to loblolly decline. Climate change, at least as it was integrated into the HexSim model, did not have an effect on population trends.

Loblolly pine decline clearly has the potential to dramatically influence RCW recovery at Fort Benning. Given the assumptions of this study's modeling approach, failure to replant longleaf pine in dead and senescing loblolly stands results in a dramatic reduction in the potential breeding pairs of RCW on the landscape (no restoration scenarios). Moreover, the interplant scenario appears to mediate the problem of loblolly decline with the least loss of potential breeding pairs over the next 100 years. The restoration scenario, which was modeled after current management on the installation, resulted in fewer RCW than the recovery goal and the interplant scenario, suggesting that the current management strategy should be revised.

A rapid decline in potential breeding pairs was observed in all management scenarios under loblolly decline in the first 10 years of the simulations. This decline likely reflects the strict senescence age imposed on a landscape with many stands of trees older than the senescence age. This rapid loss of habitat is unlikely to occur at the modeled rate, but rather will be spread-out over a larger time frame. The large reduction in habitat in the first 10 years also

leads to the peak in potential breeding pair numbers around 2075. The lowest values after the decline occur in year 2014, and the peaks occur at 2074, a 60-year gap. The habitat model includes whether the trees in a pixel are greater than or equal to 60 years of age, as nest trees must be 60 years or older (on average). The peaks in 2074 reflect the trees that were planted in 2014 becoming suitable nest trees. Thus both the rapid decline in the first 10 years and the timing of the peak for the restoration scenario are artifacts of the strictly imposed age of mortality. Despite these model artifacts, the comparison of the population trends resulting from the different management scenarios is highly informative as the relative rate of population increase after the initial decline reflects the effect of each management scenario on RCW habitat.

Two of the management scenarios (restoration and interplant) resulted in more habitat for RCWs earlier than the cut scenario, and therefore more potential breeding pairs. These results suggest that, not surprisingly, cutting large areas of RCW habitat may negatively affect RCW populations at Fort Benning for more than 60 years and may not be the best management strategy. Instead, the results suggest that interplanting longleaf pines in loblolly stands reduces the habitat loss and subsequent loss of RCW breeding pairs due to loblolly decline. Overall, the difference between the no restoration scenario (no management) and the other three management scenarios suggests that efforts to reduce the impact of loblolly decline on RCW through habitat alteration may be successful and should be undertaken. Studies are currently underway at the base to explore the feasibility of selective harvest and planting.

Simulating Land-use Change. Overall, the population trends were positive relative to the baseline (no development) scenario for all Conservation scenarios (Figure 31), slightly negative for all Convenience scenarios (Figure 32), and negative for all Worst-case scenarios (Figure 33).

Results suggest that strategies to minimize the effects of land use change on RCW at Fort Benning were successful. There were no qualitative effects of training range development under the Conservation scenarios, suggesting that careful planning could effectively reduce the impact of additional troops on RCW at Fort Benning. Indeed, even the Convenience scenarios had relatively small effects on RCW numbers. Taken together, these results suggest that future increases in training needs on the landscape at Fort Benning will not necessarily dramatically reduce the RCW population on the installation. However, it is important to note that these scenarios suggest that the current recovery goal of 351 potential breeding pairs is an unrealistic target given future development and changes in land use, as none of the scenarios approached this goal within the 100-year simulations.

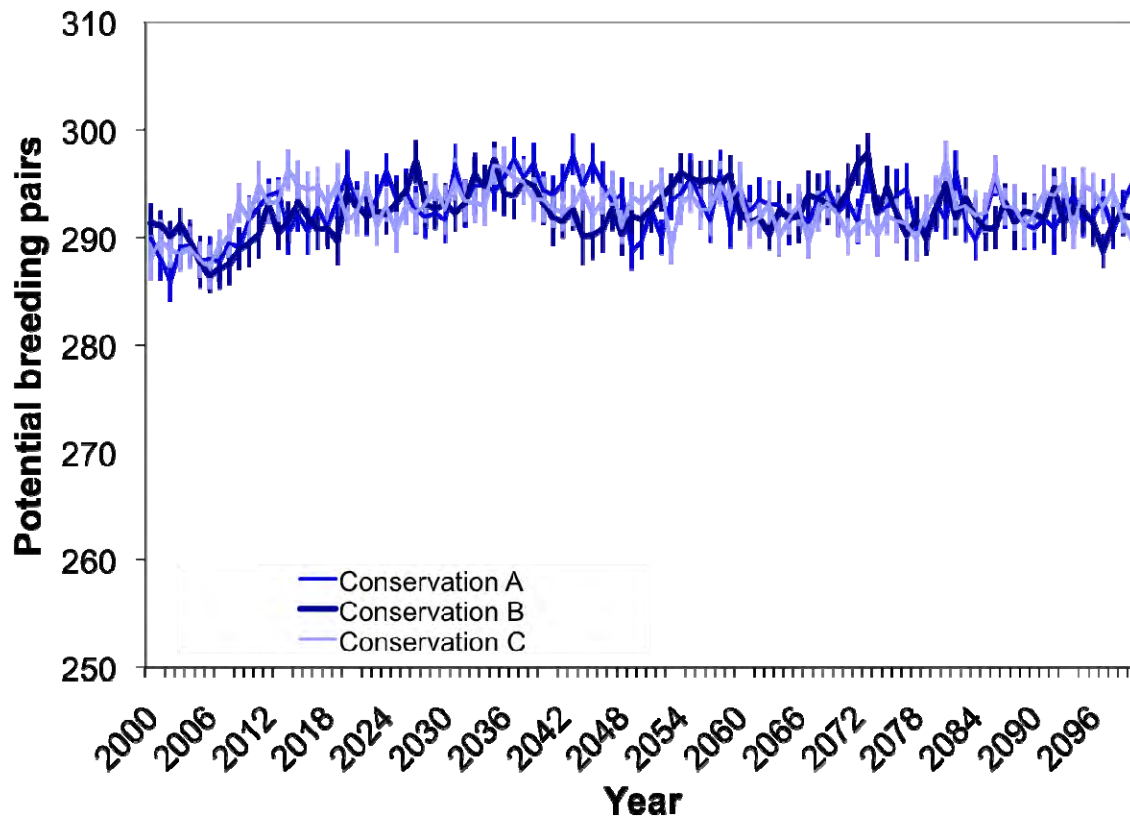


Figure 31. Red-cockaded woodpecker (RCW) population projections under Conservation land-use scenarios.

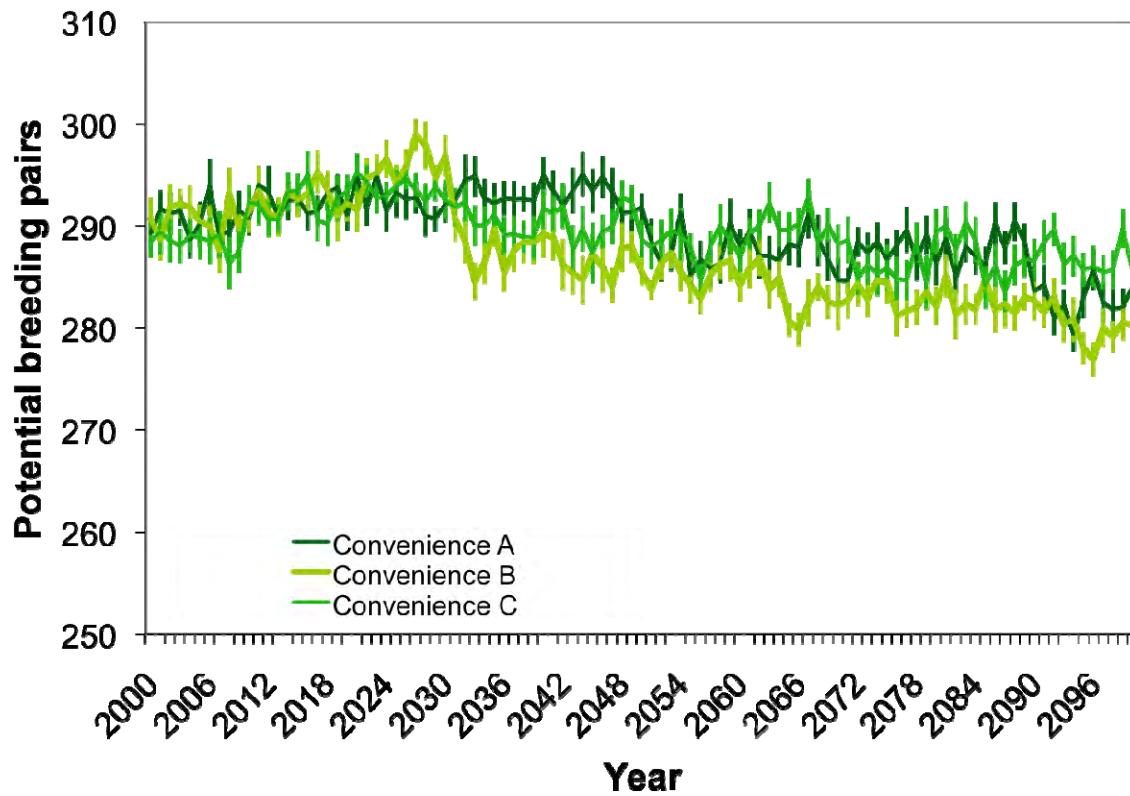


Figure 32. Red-cockaded woodpecker (RCW) population projections under Convenience land-use scenarios.

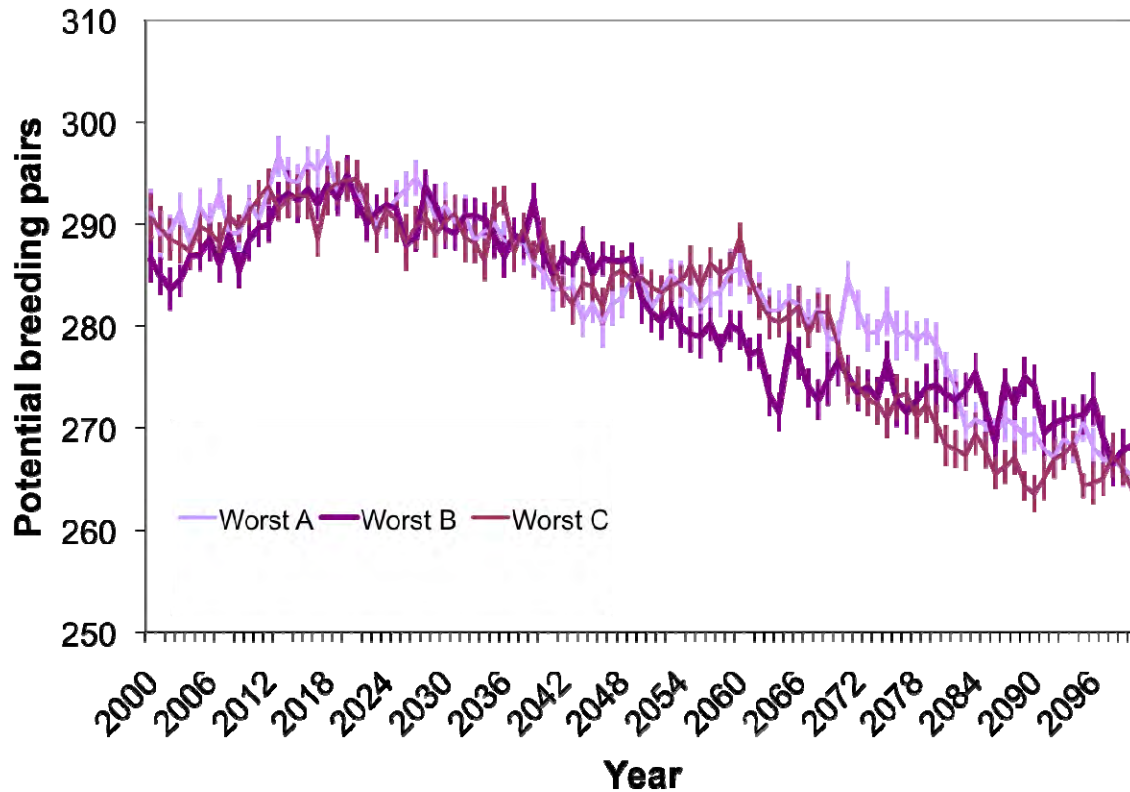


Figure 33. Red-cockaded woodpecker (RCW) population projections under Worst-case land-use scenarios.

Simulating Climate Change. Simulated climate impacts had no effect on the RCW population at Fort Benning. Population trends were not different from the baseline under any future climate scenario (Figure 34).

Climate change had very little effect on RCW population trends at Fort Benning under the current scenarios. This result was surprising given the projected increase in spring precipitation (range: +1% to +27 %) for the Fort Benning study area. However, precipitation as high as 300 mm for the month of April did not reduce the average reproductive output of any cluster below 1 fledgling. Because each cluster is still reproducing, it is possible that the population as a whole is still buffered from years of low reproduction due to the helper class of older but un-reproductive offspring. Thus, the reduced reproductive output does not scale up into the reproductive class, and no effect is seen on the population. The results suggest that RCW may be less vulnerable to climate change than other species because of the helper class. However, if mortality of breeder and helper RCW increased under climate change, the populations could be negatively affected.

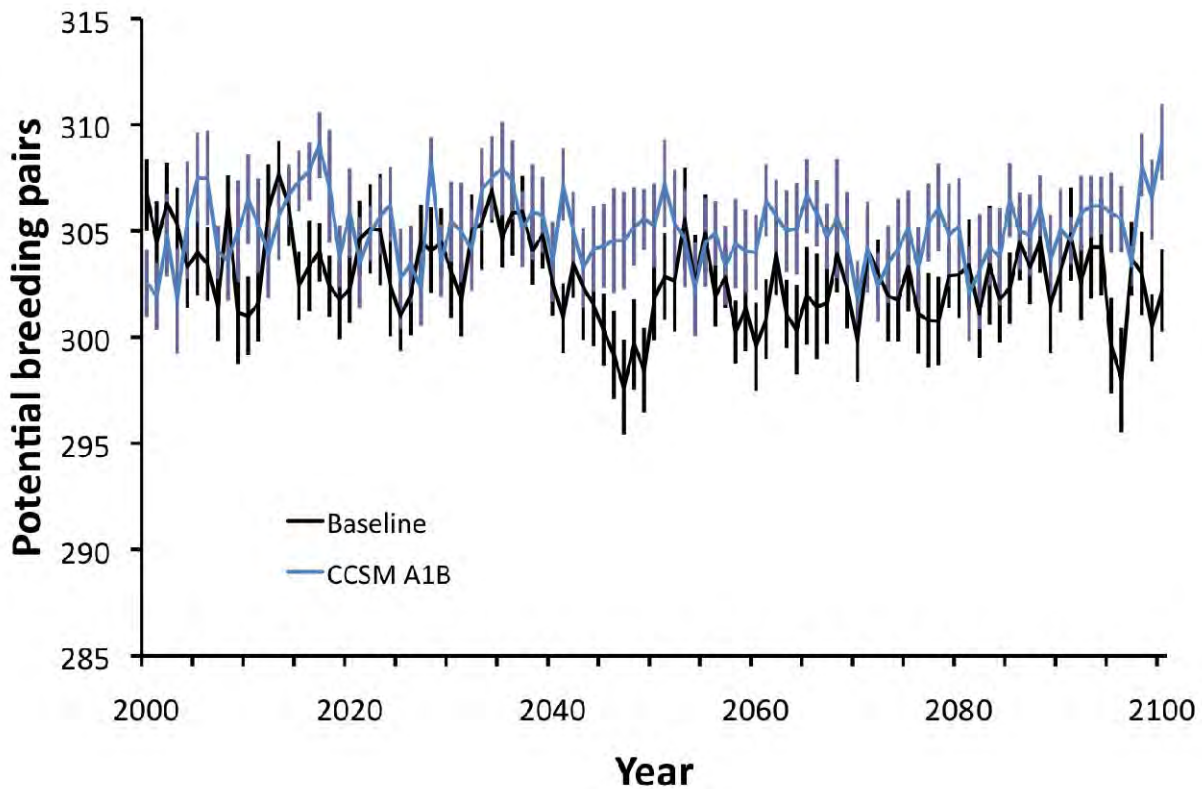


Figure 34. Red-cockaded woodpecker population projection under baseline (static precipitation) and projected future precipitation from the CCSM GCM run for the A1B emissions scenario.

Under any future conditions, it will be important for managers on the installation to continue implementing cavity insertions and hardwood understory removal. All of the simulations in the current study assumed no additional cavity installation for the RCW. Most of the observed population growth on Fort Benning in the past several years was likely driven by successful cavity creation (Doresky et al. 2003). It remains to be seen if artificial cavity creation can sustain population growth in the face of landscape changes due to loblolly decline and range development.

Black-capped Vireo case study

Vireo Habitat Model

Habitat models built with LiDAR-derived measures of vegetation structure performed well, correctly predicting between 68% and 80% of observations in the 25 test datasets (Figure 35). The relationship between LiDAR model performance and resolution was curvilinear with models at intermediate resolutions (25- and 50-m) performing better. The best performing LiDAR model used 25-m resolution data and correctly predicted a median 77% of observations.

Other measures of model performance were similar. AUC scores for LiDAR models ranged from 0.76 to 0.87 and κ values ranged from 0.33 to 0.57.

The relative importance of the four structural variables differed across resolutions. Edge density (EDGE), percent cover of vegetation 1-2m tall (COVER2), and vegetation height (HEIGHT) were all more important predictors than total percent woody cover (COVER) in the 25-m resolution model (Figure 36). COVER2 was most important in the 10-m resolution model, whereas EDGE was most important in coarser resolution models. Cover measures were positively correlated (Spearman's rank $\rho = 0.91$, p-value < 0.001) and correlated with height (COVER: $\rho = 0.79$, COVER2: $\rho = 0.66$; p-values < 0.001) at 25-m resolution. No other measures were correlated by more than $|\rho| = 0.5$.

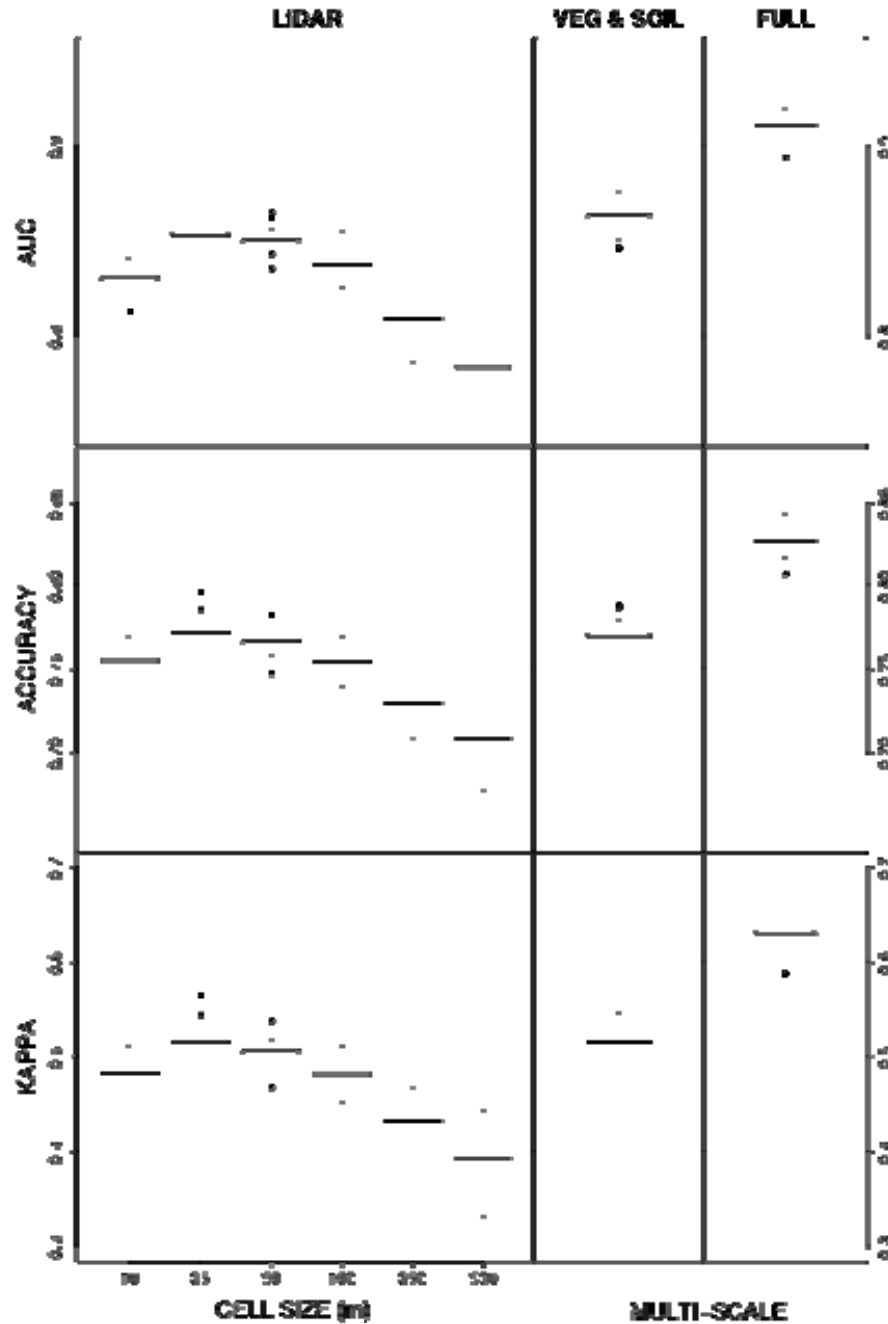


Figure 35. Threshold independent (AUC) and threshold dependent (overall accuracy and Cohen's κ) measures of model performance for models constructed with only LiDAR-derived measures of vegetation structure at six spatial resolutions (LiDAR), only vegetation and soil variables (VEG & SOIL), and including all predictor variables (FULL). The VEG & SOIL model is built with 25-m resolution presence/absence data and 75-m resolution vegetation and soil data. The FULL model includes 25-m resolution LiDAR and presence/absence data and 75-m resolution vegetation and soil data. Dashed lines mark bounds for relative measures of model fit for AUC (Swets 1988) and κ (Landis and Koch 1977) and an arbitrary minimum acceptable threshold of 75% accuracy.

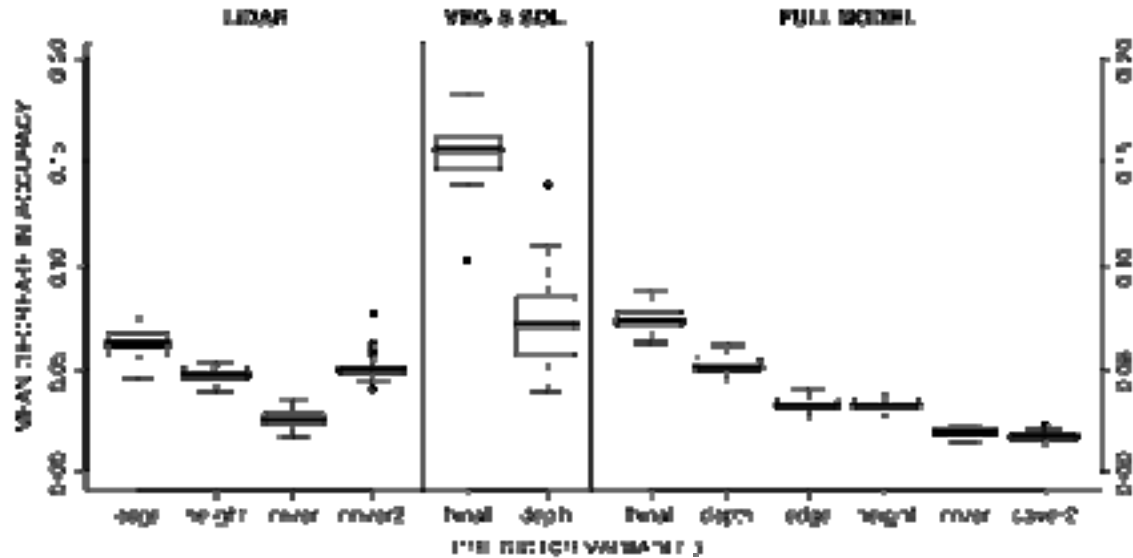


Figure 36. Mean decrease in accuracy as a measure of variable importance for each predictor variable the 25-m resolution LiDAR model and multi-scale VEG & SOIL and FULL models. Descriptions of variables and their names are provided in Table 3.

The vegetation and soil model performed similarly to the 25-m LiDAR model, correctly predicting a median 77% of observations (Figure 35). Vegetation functional type was a more important predictor than soil depth (Figure 36). The FULL model, including all LiDAR-derived measures of vegetation structure as well as vegetation composition and soil depth measures, correctly predicted a median 83% of all observations (Figure 35). Vegetation functional type was consistently the most important predictor, followed by soil depth, edge density and height, and cover measures (Figure 36).

The curvilinear response of LiDAR model performance measures to changes in data resolution (Figure 35) suggests that vireos are best associated with habitat structure summarized with 25- to 50-m resolution grid cells (0.05-0.25 hectares). Similar areas were used by Grzybowski et al. (1994) and Bailey and Thompson (2007) to calculate cover and height (0.04 ha) and percent cover and edge density (0.2 ha), respectively, in their vireo habitat suitability models. This is now the third study of vireo habitat in which these measures at this approximate spatial scale have proven important predictors of vireo habitat suitability. The 10-m grid cells may be too small to capture the combination of open grasslands and patchy shrublands associated with vireos, resulting in poor performance of the finer resolution models. Aggregation to coarser resolutions introduces errors (Bian and Butler 1999) that likely decrease model performance.

Models based on LiDAR-derived measures of vegetation structure performed similarly to models based on vegetation functional type and soil. Combining all measures in a single multi-scale model increased median accuracy from 77 to 83%. Incorporating data from multiple spatial scales to improve model performance is a relatively new but increasingly applied technique. Habitat suitability models can be built directly from data summarized a multiple spatial scales as we did (also see Graf 2009), or single scale models can be combined in a model ensemble. Model ensembles may outperform single-scale models when the single-scale models are not built with techniques such as Random Forest or cforest that are themselves ensembling

methods (Bancroft et al.). Our approach was different because unique vegetation characteristics were summarized at each spatial scale. This likely contributed the increased performance of the FULL model more than the combination of spatial scales.

Despite producing similar mean accuracies, the LiDAR and VEG & SOIL models provide different, yet complementary, predictions of habitat suitability across Fort Hood (Figure 37). The LiDAR model identifies suitable habitat structure throughout Fort Hood, including some regions where very few vireos were observed. These may be regions with suitable structure, but dominated by juniper or shrub species avoided by vireos. Similarly, the VEG & SOIL model identifies large patches of suitable habitat, particularly in southeastern Fort Hood, where few vireos have been documented. This area has the correct species composition but is currently overgrown. The FULL model balances both measures, revealing the benefit of including both elements in a model.

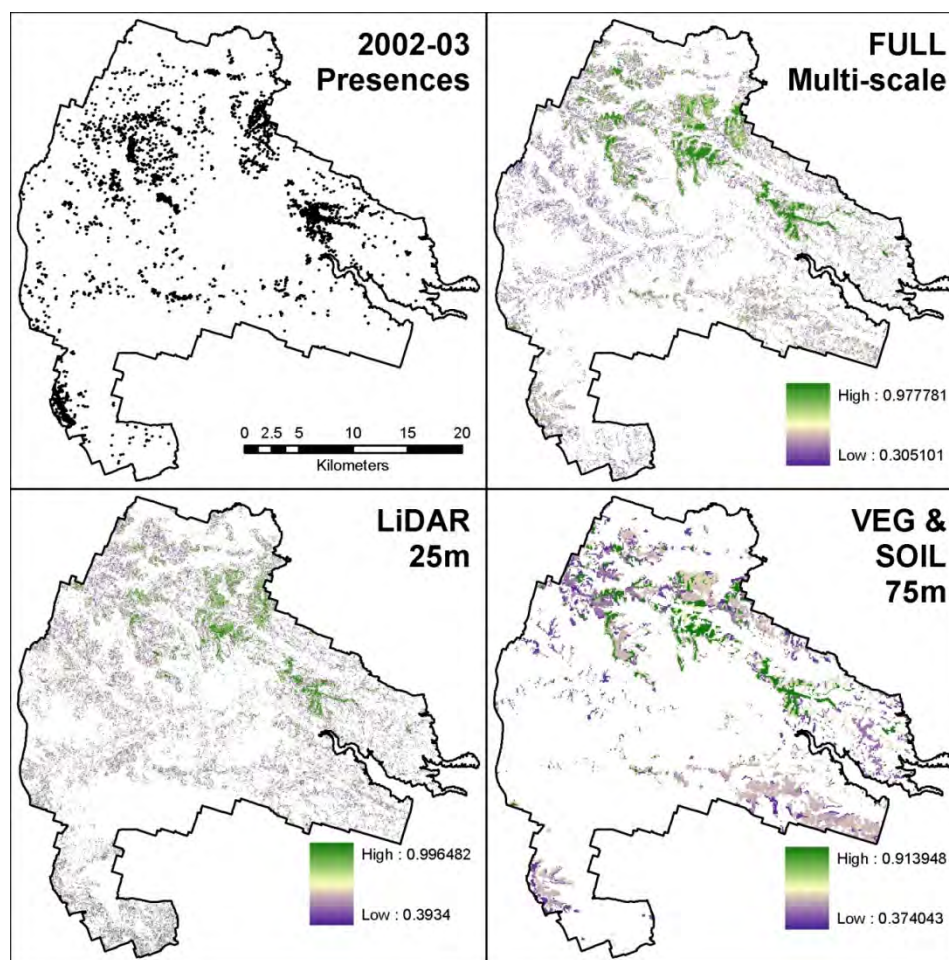


Figure 37. Black-capped Vireo presence/absence data and predicted probability of presence based on suitability models constructed with 25-m resolution LiDAR-derived measures of vegetation structure (LiDAR), 25-m resolution presence/absence data and 75-m resolution measures of vegetation functional type and soil depth (VEG & SOIL), and all variables combined (FULL).

The best models performed well in both threshold independent (AUC) and threshold dependent (κ and overall accuracy) measures of model fit. Median AUC values for our models were equal to or higher than reported values in other avian habitat suitability modeling efforts using LiDAR-derived measures with (Graf et al. 2009) and without (Seavy et al. 2009) measures of vegetation composition. Median κ values were higher than values reported for a LiDAR-derived model of grouse habitat (Graf et al. 2009). Median accuracies were above 75% for a majority of models. This level of accuracy would be acceptable for most habitat management applications. However, managers may choose to select a more conservative threshold (one that minimizes commission error) than we used for designation of critical habitat.

The age and asynchrony of the data sources used likely introduced additional errors into our models. Vegetation growth, disturbance, and annual variation in the presence/absence of vireos throughout Fort Hood would likely explain much of this error. In spite of this, models performed well, suggesting that broader patterns of vegetation structure, composition, and vireo occupancy remain consistent through time. However, this model as well as others employing remotely sensed data should be validated in the field with current presence/absence data before being applied in a management context.

Simulating Habitat Change

Model Output

Figure 38 includes examples of outputs from a single model run of the FHVM. Maps reveal changes in the distribution of vireo habitat over time resulting from the implementation of training and fire disturbances. Output from the model is variable even within the same scenario because each stochastic run results in a unique number of fires per year, fire extents, and fire locations. Coefficients of variation for the 30 model runs completed for each scenario ranged from 11-21%.

Predictions of total vireo habitat area in 2060 were similar among scenarios (Figure 39). The Large Fires scenario increased the median extent of projected habitat relative to the Baseline and two Training scenarios, whereas the Fire Suppression scenario decreased habitat availability. Training scenarios resulted in no change in the median projected vireo habitat area in 2060. Only a handful of model runs resulted in predicted habitat extents greater than the current extent of vireo habitat. Median predictions were roughly 15% less than current habitat totals. Results were similar among scenarios when only the extent of habitat with a probability of vireo occurrence greater than 0.95 was compared (Figure 39). These grid cells have the highest probability of vireo occurrence and can be viewed as core habitat. Projected extents of core habitat were more similar to current extents.

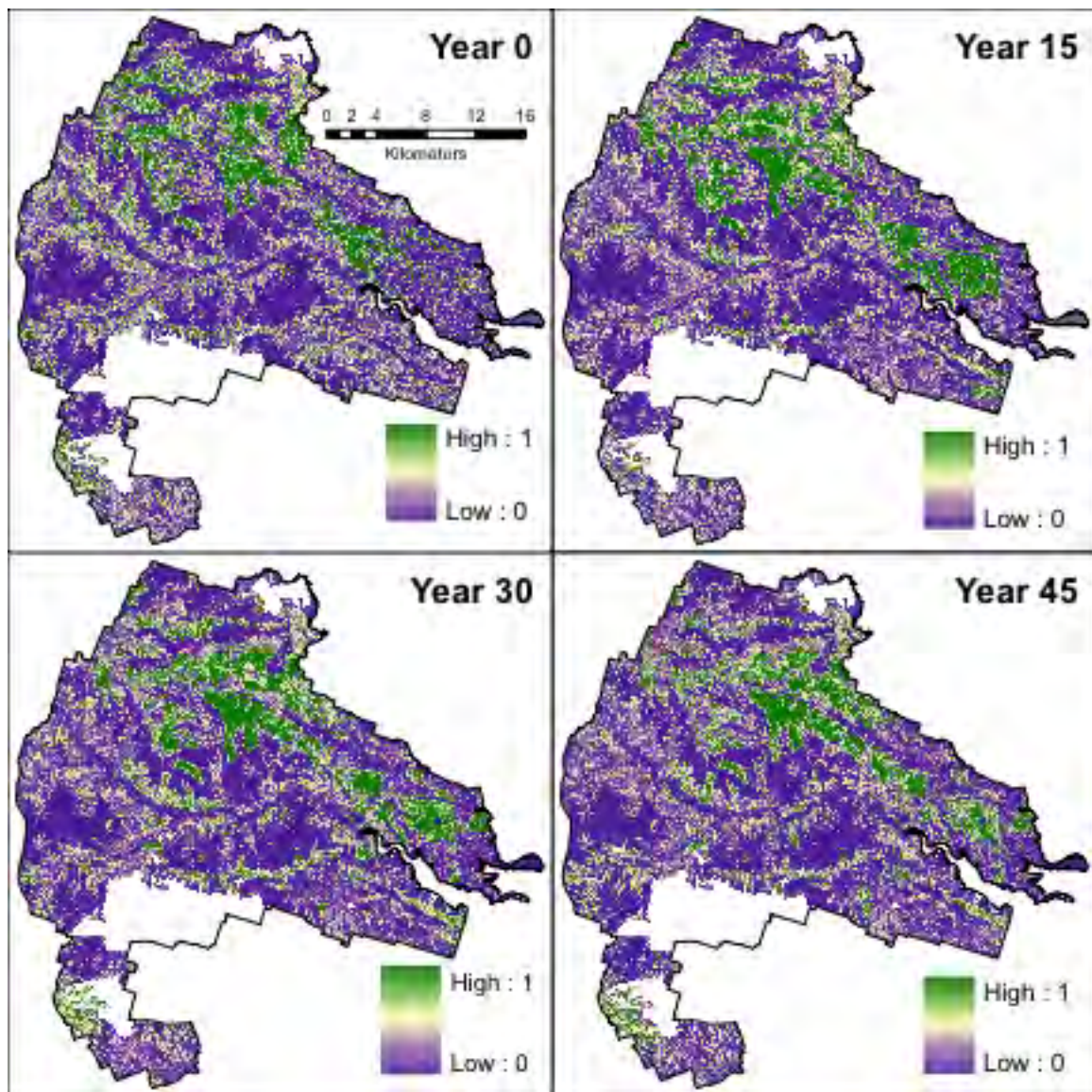


Figure 38. Current vireo habitat quality (year 0) and output from the Fort Hood Vegetation Growth and Disturbance model for simulated years 15, 30, and 45. Output is from a single run of the Larger Fires scenario.

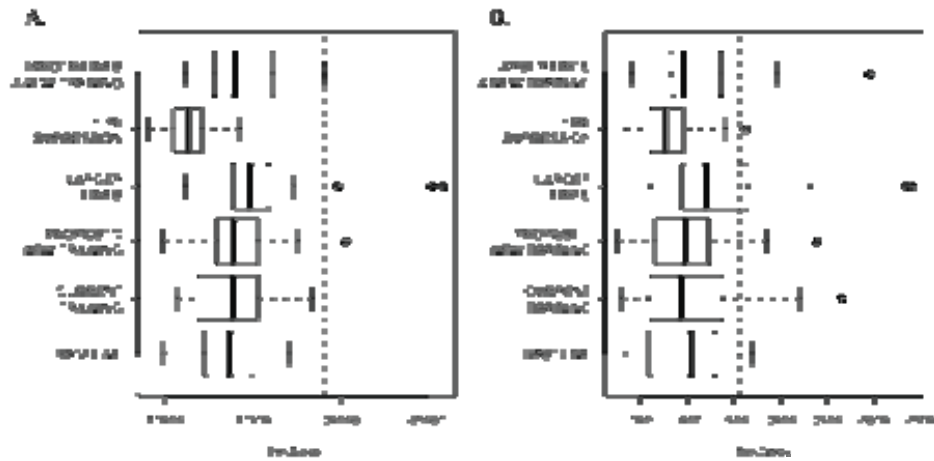


Figure 39. (a) Boxplots of predicted total area of vireo habitat in 2060 from 30 simulation runs for each of six scenarios. (b) Boxplots of predicted area of vireo habitat with a probability of vireo occurrence greater than 0.95. Vertical dashed lines mark the present estimates of suitable vireo habitat from the Vireo Habitat Suitability Model.

Sensitivity Analysis

The simulated extent of suitable vireo habitat after 30 years was most sensitive to the size and number of fires, vegetation height growth rates, and the cutoff used in designating suitable habitat in the Vireo Habitat Suitability Model (Table 12). The cutoff in the habitat model had the largest coefficient in the linear regression model based on 500 Monte Carlo runs, followed by the two parameters defining the extent of modeled fires. The adjusted R^2 value for the linear model fit to the Monte Carlo dataset was 0.327.

Table 12. Parameters, their potential ranges, and sensitivity coefficients for the Fort Hood Vegetation Growth and Disturbance Model. Sensitivity coefficients were estimated from 500 model runs using multiple linear regression.

Parameter	Description	Value	Coefficient	p-value
Cutoff	Threshold in the Vireo Habitat Suitability Model for designating habitat/non-habitat	0.369	-36355	<0.0001
Lambda	Parameter in the Poisson distribution sampled to determine the number of annual fires	5	8910	0.0087
Dry Year Probability	Probability (p) in the binomial trial for selecting high-fire years	0.2	3853	0.225
Dry Factor	Multiple used for increasing the number of fires in high-fire years	10	-990	0.759
Fire Size (Shape)	Shape parameter for the gamma distribution sampled to determine fire extent	7.304	13001	<0.0001
Fire Size (Scale)	Scale parameter in the gamma distribution sampled to determine fire extent	0.503	21874	<0.0001
K_1	Parameter in the edge density growth function	35.44*	-3388	0.314
K_2		26.95*	4863	0.153
K_3		-0.0783*	1738	0.597
K_m		29.34*	7060	0.032

* Representative parameter value. Proportional relationships among parameters for models of different vegetation types or soil depths were held constant during the Monte Carlo sensitivity analysis.

The median projected vireo habitat availability was indistinguishable among the baseline and military training scenarios. It was expected that military training would impact projected habitat, but the observed absence of an effect reinforces the emerging consensus that military training activities are benign to vireo habitats (D. Cimprich, pers. comm. USFWS 2005). These results suggest that opening additional training areas on the east range is unlikely to have a severe, negative impact on vireo habitats.

Climate-induced increases in the annual area burned resulted in increased vireo habitat availability, whereas fire suppression led to declines. The Larger Fires scenario represented a conservative estimate of increased fire activity. Greater change is likely and would probably increase projections of available vireo habitat. These results suggest that large fires, such as the 1996 fire, are necessary for the creation and maintenance of vireo habitat. In the absence of these events, the area of available vireo habitat will decline, as occurred in the Fire Suppression Scenario. Fire suppression also decreased the availability of core habitats.

Military training may offset the increase in vireo habitat availability resulting from increased fire. This effect, observed in the Train West and East Ranges and Larger Fires scenario, appears to result from the thinning component of the training module. Thinning in the east range resulted in a decline in habitat availability that offset some of the gains from increases in the area burned. There is no evidence that this interaction is synergistic.

Output from the FHVM suggests that the current disturbance regime would maintain vireo habitat availability lower than current its extent over the long term. The Fort Hood vegetation biologist concurs and considers current extent of vireo habitat to be high relative to historical conditions (C. Reemts, personal communication). Therefore, output from the FHVM

is consistent with what are believed to be historical conditions, whereas the present extent of vireo habitat was strongly influenced by the 1996 fire. Results from the Larger Fires scenario also suggest that additional habitat would result from an increase in the frequency of large fires.

Parameter Sensitivity and Uncertainty

Predictions of the future extent of vireo habitat produced by the FHVM were sensitive to parameters impacting the size and number of fires. Parameters impacting the size and number of fires were derived empirically using the Fort Hood fire database. This database was begun in 1994, but is not considered exhaustive. Therefore, the database likely underestimates fire activity. Observed fires during this period were also influenced by fire management. Fort Hood enforced an “extinguish all” policy from 1996-2004 and a “let burn” policy from 2004-present (Pekins 2006). This shift in management practice likely underestimates fire activity from 1996-2004 and over-estimates activity, due to the accumulation of fuels, from 2004 until the present. These potential sources of error would directly impact estimates of fires size and the number of fires per year. Furthermore, the ratio of fires occurring within and outside of the LF area may also be influenced by these changes, although model output was not sensitive to this parameter.

Output from the FHVM was also sensitive to the initial rate of shrub growth following disturbance. Species-specific data was not available to characterize shrub growth and the best available data was used. The post-mulching growth rates included multiple species and soil types, capturing the range of possible growth rates, but these data were pooled in our parameterization. Differences in growth models among vegetation types were determined entirely by the LiDAR-derived estimate of maximum height. Therefore modeled growth rates were similar among species, especially in the first five years post-fire. Vegetation recovery following the 1996 wildfire generally supports a homogeneous and rapid response: vegetation returned to the pre-fire community structure and stem densities, absent juniper, within ten years (Reemts and Hansen 2008).

Uncertainty in the Vireo Habitat Suitability Model strongly affected output from the FHVM. The classification model assigns grid cells a probability of vireo occurrence. A cutoff probability was selected using a test dataset and the criteria that errors of omission and commission were made equal. This criterion was selected to reduce bias. However, management context may dictate the need for a more conservative threshold that minimizes commission errors. This would lead to a subsequent reduction in the predicted extent of vireo habitat from the FHVM. Asynchrony in the datasets used to construct the Vireo Habitat Suitability Model may also introduce errors in the projected extent of habitat suitability. The last installation-wide survey for vireos was conducted in 2002-2003. These data were used in order to construct a habitat suitability model for all of Fort Hood. However, LiDAR-derived measures of vegetation height and edge density were collected in 2009. The six years separating these two surveys likely inflates the range of heights and edge densities considered suitable for vireos. This would lead to habitat output from the FHVM being scored suitable when it is already too tall, resulting in a bias towards over-estimating the extent of projected vireo habitat.

Output from the sensitivity analysis should be qualified due to the low coefficient of determination (R^2) value for the multivariate linear regression model (Saltelli 2004). The relationship between predictor variables and model output is likely non-linear, including interactions between input parameters. This detracts from the “global” quality of the sensitivity analysis; however, the Monte Carlo approach does explore more parameter combinations than a one-at-a-time local sensitivity analysis (van Nes et al. 2002, Saltelli 2004).

In conclusion, bias introduced into the FHVM due to input data likely led to an under-estimation of projected vireo habitat availability and baseline disturbance rates. A smaller window for habitat suitability, slower vegetation growth, and more fires would increase projected habitat availability, moving projections closer to current levels of habitat availability.

Simulating Population Responses to Multiple Stressors

Model Calibration.

The VCPM was able to reproduce both the observed rapid growth of the vireo population during the period of base-wide cowbird control and the decline following the cessation of control on the west range (Figure 40). When base-wide cowbird control is sustained until year 40 in the model, the vireo population maintains a stable abundance within the range of vireo abundances observed in 2005 and 2006 (Figure 40a). The VCPM was unable to recreate the 52% decline in estimated vireo abundance between 2006 and 2007; five years were required for the modeled vireo population to fall within the range of population estimates from 2009 and 2010 (Figure 40b).

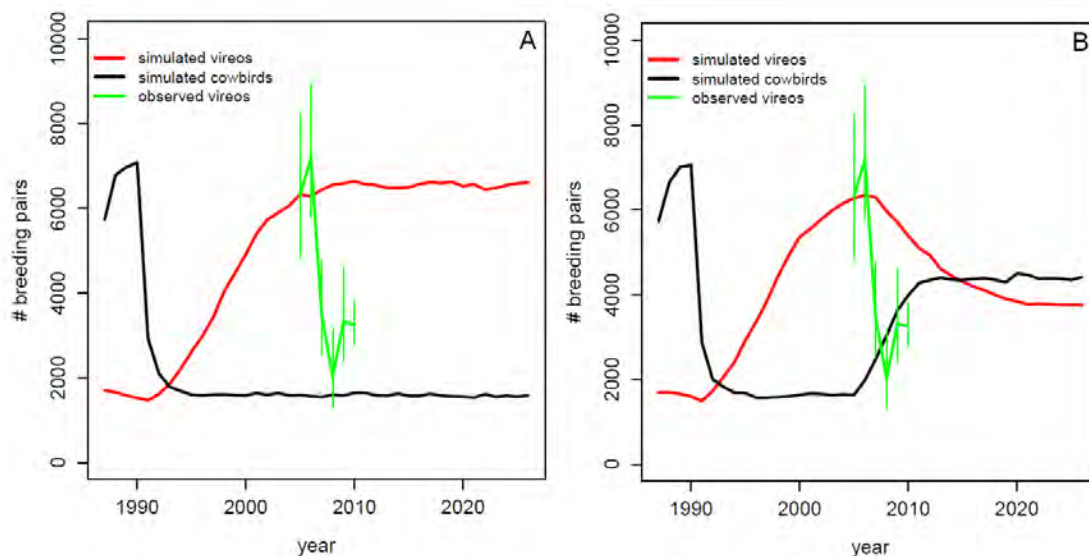


Figure 40. Modeled vireo and cowbird abundances in the calibrated Vireo-Cowbird Population Model. The modeled vireo population increases rapidly in response to base-wide cowbird control beginning in 1990, reaching a plateau in 2005 that falls within the confidence limits of the observed vireo population estimates for 2005 and 2006 (A). When modeled cowbird trapping on the west range is stopped in 2005 (B), the modeled vireo population decreases to an equilibrium abundance that falls within the confidence limits of the observed vireo population estimates for 2009 and 2010.

Sensitivity Analysis.

Output from the VCPM is sensitive to a number of model parameters for both populations, some of which interact strongly with each other. Two of the eight parameters for the vireo population sub-model demonstrated relatively high direct impacts on modeled vireo

abundance (μ^*) and effect of interactions with other parameters (σ): vireo productivity (bcPro) and juvenile survival (bcJuvSurv, Figure 41a). The proportion of total productivity modeled in a parasitized nest (bcProP) and the percent of the vireo population susceptible to parasitism (bcParSus) also had moderate effects on modeled vireo abundance. None of the cowbird parameters had a notable effect on modeled vireo populations; however, measures of interaction-effects (σ) for vireo parameters were relatively higher than for cowbird parameters (Figure 41b).

Cowbird survival out of the range of traps (bhOutSurv) had the greatest direct impact on modeled cowbird populations as well as higher indirect effects (Figure 41b). Five other parameters had moderate effects: range of trap effectiveness (bhThresh), minimum resource requirements for cowbirds to establish territories (bhRes), cowbird territory size (bhHa), cowbird productivity (bhPro), and cowbird survival within range of traps (bhInSurv, Figure 41b). Again, none of the parameters for the vireo sub-model influenced cowbird populations as would be expected since there is only a one-way interaction between populations.

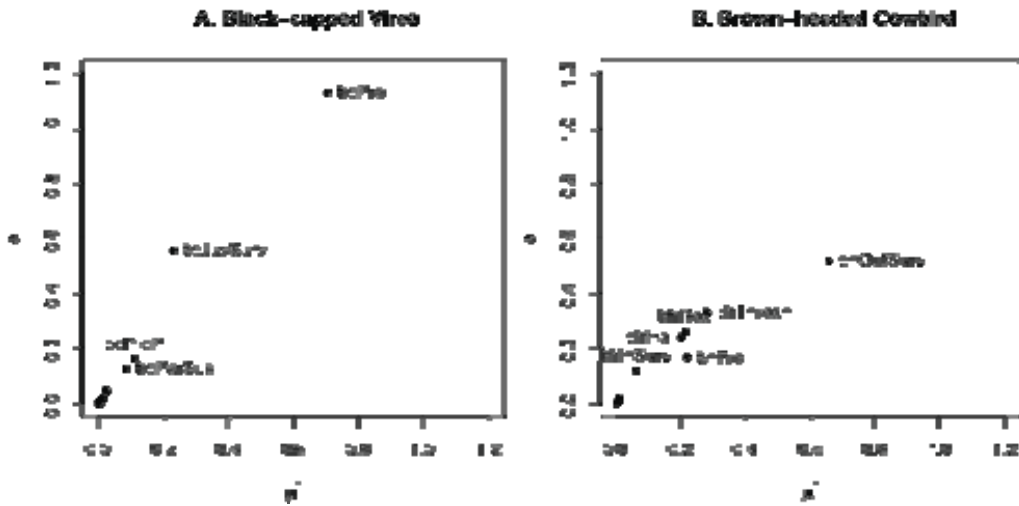


Figure 41. Measures of model parameter's direct (μ^*) and interaction effects (σ) on the number of modeled vireos (a) and cowbirds (b) from the Vireo-cowbird Population Model based on the Morris method for sensitivity analysis. Parameter codes: bcPro = vireo productivity, bcJuvSurv = vireo juvenile survival, bcProP = proportion of total vireo productivity assigned to a parasitized nest, bcParSus = percent of vireos susceptible to parasitism by cowbirds, bhOutSurv = cowbird survival out of trap range, bhThresh = range of trap effectiveness, bhRes= minimum resources required for a simulated cowbird to establish a breeding territory, bhHa = cowbird target territory extent, bhPro= cowbird productivity, bhInSurv = cowbird survival in range of traps.

Model Output

Projected vireo abundances differed between military training and fire scenarios. Mean abundances based on moving averages were highest in scenarios with climate-induced increases in fire extent and lowest in the fire suppression scenario (Figure 42a). Baseline and military training scenarios had intermediate projections of mean vireo abundance. Without cowbird control, differences among scenarios were erased as all simulated populations declined to ~1200

breeding pairs. Therefore, the impact of decreased cowbird control strongly overshadows simulated differences in military training, and fire scenarios.

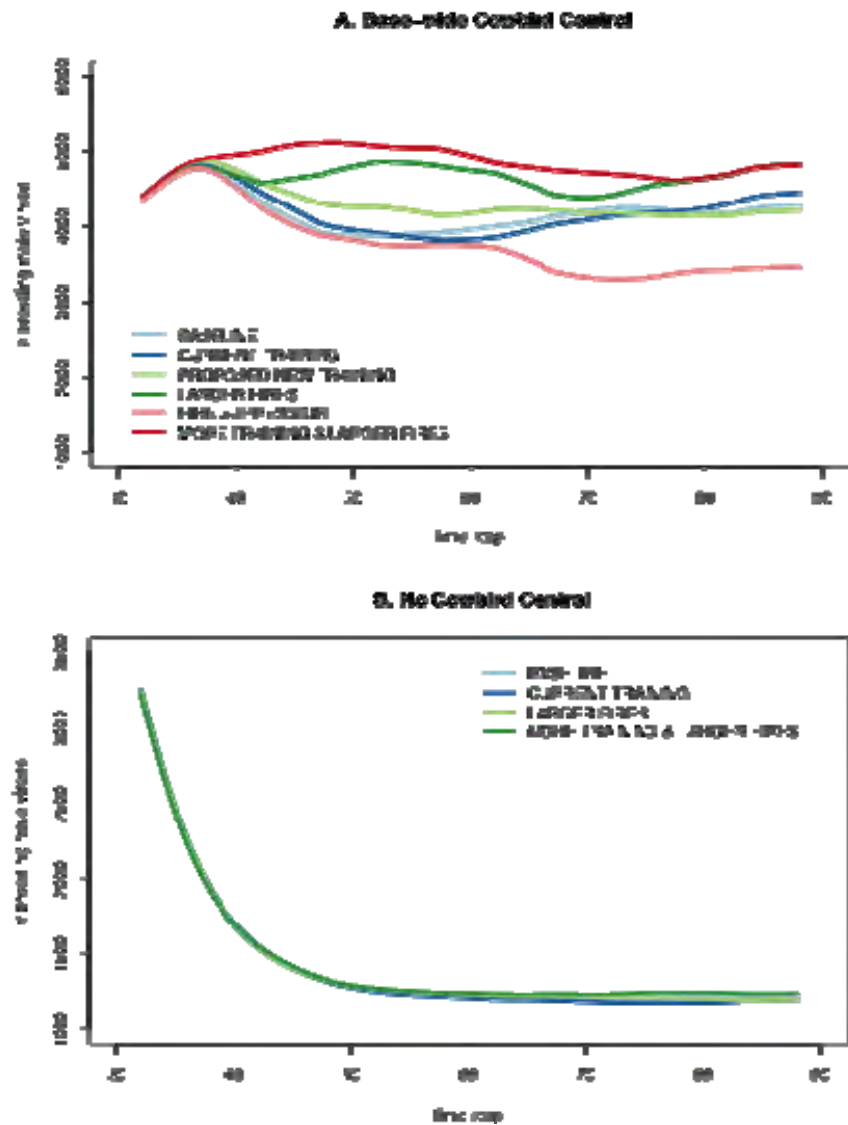


Figure 42. Moving averages of modeled vireo populations from the Vireo-Cowbird Population Model. Each scenario is based on 75 model runs from 15 vegetation projections output from the Fort Hood Vegetation Growth and Disturbance Model. Population model run with (a) and without (b) base-wide cowbird control.

Projections for the climate-induced increase in fire extent scenario resulted in greater intra-scenario variability in projected vireo abundance than other scenarios (Figure 43). This variation originates from vegetation projections more so than from stochasticity in VCPM vital rates. Coefficients of variation for output from the Fort Hood Vegetation Growth and Disturbance Model ranged from 11-21%, whereas the coefficients of variation from repeat runs

of the VCPM were ~1%. Median projected abundances from all scenarios fail to surpass the maximum vireo population estimate from 2006.

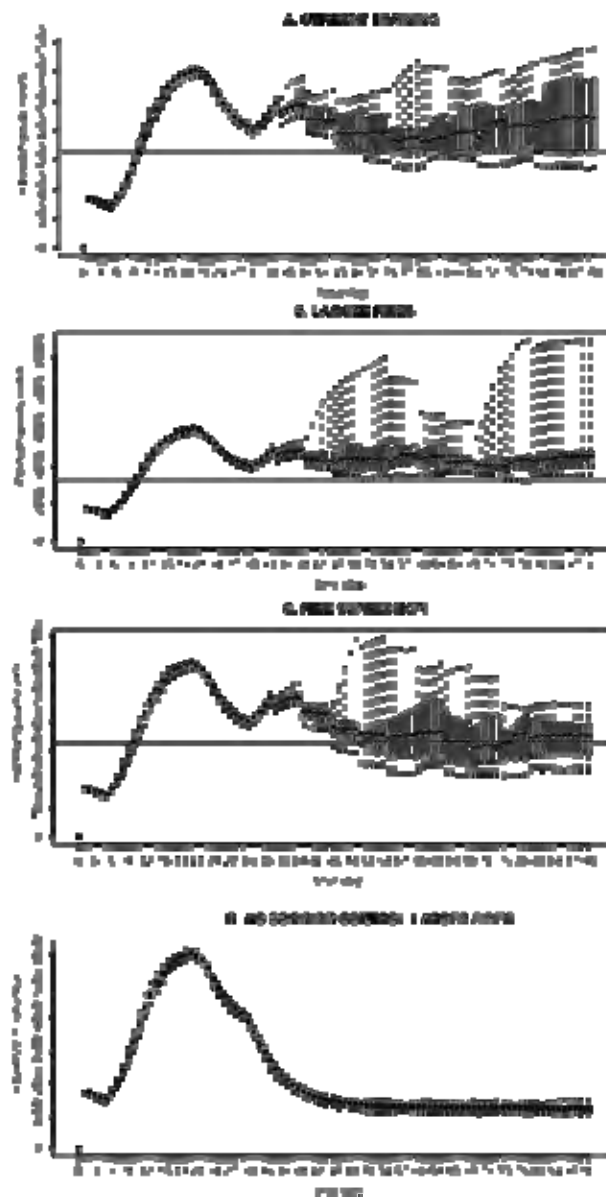


Figure 43. Annual projected vireo population abundances summarized with boxplots displaying the median and full range of predicted vireo abundances across all 75 model runs from 15 vegetation projections. Current Training (a), Larger Fires (b), Fire Suppression (c), and Larger Fires without Cowbird Control (d) scenarios are represented. Horizontal line marks the 2010 vireo abundance estimate for Fort Hood.

Simulated populations from all scenarios maintained abundances above the minimum 1000 pairs mandated by the Fort Hood Endangered Species Management Plan (USFWS 2005).

All scenarios except Fire Suppression had a greater than 0.5 probability of maintaining a population greater than 3000 breeding pairs (Table 13). The Baseline and Train West & East Ranges and Larger Fires scenarios had the highest probabilities of maintaining populations above 4000 individuals. Only the combined scenario had a negligible probability of maintaining a population above 5000 breeding pairs. All probabilities were drawn from output of the VCPM pooled for each of the 15 runs of the Fort Hood Vegetation Growth and Disturbance Model, because variation due to the VCPM was minimal.

Table 13. Probability of a scenario maintaining a given minimum population in the next 50 years (n=15 vegetation model runs).

Scenario	>2000	>3000	>4000	>5000	>6000
Baseline	1	0.6	0.2	0	0
Train West Range	1	0.8	0	0	0
Open East Range	1	0.733	0.067	0	0
Larger Fires	1	1	0.133	0	0
Fire Suppression	1	0.4	0	0	0
Train West & East Ranges and Larger Fires	1	0.8	0.333	0.133	0

Scenario Effects

Results from the VCPM emphasize the need for sustained cowbird control activities to maintain current vireo abundances on Fort Hood. Simulated increases in disturbance and habitat due to increased military training and climate-induced changes in area burned did not offset the direct demographic impact of increased cowbird brood parasitism on vireo productivity. The simulated number of breeding pairs in the no control scenarios should not be interpreted as point estimates of the minimum sustained vireo population in the absence of cowbird control. The VCPM was parameterized using population estimates from 2005-2010 and therefore should not be used to characterize population behavior under maximum rates of parasitism. Reported estimates of vireo abundances prior to initiation of cowbird control in 1990 were as low as 87 individuals (Tazik et al. 1993). These estimates are considered conservative because they were based on direct observation, but vireo abundances below the modeled ~1200 breeding pairs are possible.

The recovery goal for Fort Hood vireo population is 1000 breeding pairs (USFWS 2005). All explored scenarios for fire management and climate-induced changes in fire extent are likely to maintain sufficient habitats for the next 50 years to satisfy this minimum requirement. In addition, results suggest that fire could have long-term impacts on the availability of vireo habitats on Fort Hood. Fire suppression outside the Live Fire Area will likely reduce the availability of vireo breeding habitat. Therefore, the “let burn” policy initiated in 2004 should be sustained. This could have negative impacts on habitat availability for the Golden-cheeked Warbler. The current policy maintains a network of fire breaks across the installation for fire suppression in the event that warbler habitat is in danger. Yet, simulations suggest that maintaining current population levels requires large wildfires, like the 1996 fire which also destroyed 2108 ha of warbler habitat (Goering 1998), to occur. The 30% increase in fire extent modeled in the Fort Hood Vegetation Growth and Disturbance Model is conservative compared to projections for other regions of North America (Flannigan et al. 2005). Therefore, impacts of wildfire on vireo habitats may be much larger and associated risk to warbler habitats much

larger. Prescribed fire could be used to maintain shrubland habitats, but to date this practice has been implemented only a handful of times because shrubland prescribed fires require ideal conditions to avoid fire escape (C. Reemts, personal communication). Mechanical removal for juniper and other vegetation from overgrown shrublands has been presented as a method for maintaining vireo habitats without fire. However, this is expensive and impractical for use over large areas. Therefore, while the current “let burn” policy for vireo habitats and “extinguish” policy for warbler habitats could reflect an appropriate balance, the warbler is at greater risk than the vireo of losing habitat to climate-induced increases area burned.

Increased military training activities as modeled are unlikely to have a strong negative impact on vireo abundances at Fort Hood. Training restrictions in designated core vireo breeding habitats were lifted in 2006 and to date no negative impacts on vireo nest depredation rates, daily survival, or overall success has been recorded (Cimprich and Comolli 2010). Therefore, it is reasonable that modeled changes would have little impact.

Parameter Sensitivity

Modeled vireo abundance displayed strong sensitivity to parameters influencing population growth during periods of cowbird control, including vireo productivity, juvenile survival, productivity of parasitized territories, and the proportion of vireos susceptible to parasitism, all parameters that would influence the population-level response to falling nest parasitism. Productivity was modeled as a bimodal distribution in which the expected value (see Table 8) was equivalent to the maximum observed mean productivity per territory. Other parameterizations were tested but discarded because vireo population under base-wide cowbird control was too slow. Juvenile survival is generally difficult to assess due to tendency of juveniles to disperse away from their natal site (Kostecke and Cimprich 2008). Adult and juvenile survival in the VCPM model (0.44 and 0.6, respectively) were similar to estimates derived from re-sighting and mark-recapture data collected on Fort Hood (0.42 and 0.54 Kostecke and Cimprich 2008). No estimates of the productivity of parasitized territories or the proportion of vireo susceptible to parasitism exist. These parameters were calibrated based on model fit to observed abundances and are subject to the uncertainty associated with the parameterization of any complex mechanistic model, namely that multiple parameter sets may produce the same observed pattern (Wimberly and Kennedy 2008). See below for further discussion of model calibration.

Parameters associated with survival, trap effectiveness, and territory size had the greatest impact on modeled cowbird density. Estimates of cowbird parameters were based on published data when available, but otherwise the cowbird model was calibrated indirectly using vireo data. Records of the number of cowbirds trapped per unit effort do exist (Kostecke et al. 2005), but these include an unknown and substantial proportion of migratory cowbirds that do not breed at Fort Hood. Uncertainty in the parameterization of a cowbird model could bias modeled effects of cowbird control, particularly effects on spatial patterns. For example, the distance to which cowbird traps are effective could affect local cowbird extirpation and parasitism rates. These biases could influence modeled rates of population growth and decline in response to trapping and may alter long-term equilibrium population estimates. Incorporating cowbirds into ongoing vireo point counts could provide some data with which to better parameterize the cowbird sub-model of the VCPM.

Model Deficiencies

Model calibration was based on criteria for population growth and decline in response to cowbird trapping. However, the VCPM was not able to replicate the precipitous decline in vireo abundance between 2006 and 2007. This deficiency suggests that either existing mechanisms are poorly defined in the model or that the observed decline resulted from mechanisms absent in the model. Substantial effort was put into recreating the observed pattern with the current model structure. Preliminary analyses incorporating environmental stochasticity into annual estimates of vital rates suggest that environmental variability would have had to be very high to recreate the observed decline. Kostecke and Cimprich (2008) estimated process error associated with vireo survival at ± 0.7 , but this magnitude did not alter model output substantially. Introduced variation in productivity may have a greater effect.

When cowbird parasitism was low, nest depredation by other species was the primary cause of nest failure (Cimprich 2009). Snakes and fire ants were the primary nest predators observed at Fort Hood (Stake and Cimprich 2003). Excluding nest predation from the model assumes that its effects are constant and integral to productivity and survival rates. This may be a reasonable assumption; however, nest depredation rates vary spatially at Fort Hood (Cimprich 2009) and may therefore interact with cowbird control efforts and the experimental cessation of controls on the west range. Currently nest predation rates are known for the six intensive study areas. Nest predation could be incorporated into the VCPM if factors influencing predation rates could be modeled across Fort Hood.

Likelihood-based Methods of Model Calibration

Calibration methods employed for the VCPM could be made more robust. Calibration is made more difficult by the computational cost of the running the VCPM; a single run takes more than 20 minutes. This limits the suitability of grid-search methods that require tens of thousands of model runs. Gradient search algorithms can be combined with likelihood-based objective functions to explore a limited portion of parameter space and optimize model fit (Hilborn and Mangel 1997, Burnham and Anderson 2002). Because the VCPM is spatially explicit, model fit should be assessed against multiple spatially-disaggregated datasets in a multi-criteria model assessment (Grimm et al. 2005, Komuro et al. 2006).

Data from each of six intensive study areas could be used for a spatially explicit model validation. The probability of a given parameter set (θ) given observed data is proportional to the likelihood as calculated with a likelihood function. Observation error, such as that associated with observed territory density can be estimated with an over-dispersed negative binomial distribution. The likelihood of the observed parasitism rates given parameters θ and assuming a negative binomial observation error would be

$$L(data_{abundance,replicates} | \theta) = \prod_{years} \Pr \{Z = (observed_{year} - simulated_{year})mu, n\}$$

The parameters mu and n are estimates of the mean error and variation in error, respectively. Likelihoods from multiple stochastic replicates of the same parameter set are averaged

$$L(data_{abundance} | \theta) = \frac{1}{n} \sum_{replicates}^N L(data_{parasitism,replicates} | \theta)$$

The final likelihood for θ is the product of likelihoods for each data type, such as abundance, parasitism frequency, productivity, etc.

$$L(alldata | \theta) = \prod_{datasets} L(data_x | \theta)$$

Weights can be applied to likelihoods from multiple data sources in the final likelihood to reflect a preference that the model fit a particular pattern over another. Alternatively, likelihoods may be kept separate, model fit scored on a binary scale, and parameter sets evaluated on their ability to fit multiple patterns.

This likelihood-based approach has been explored with the VCPM attempting to fit model output to both base-wide estimates of vireo abundance as reported above and site-specific counts of vireo nesting territories at six long-term nesting sites (Cimprich, 2010). Exploratory analyses revealed the challenge of fitting both fine and broad scale data simultaneously. One problem results from the depiction of vireo habitat. The habitat suitability model currently used in conjunction with the VCPM is calibrated to presence/absence data. Therefore, it does not capture variation in habitat quality among occupied sites. Modeled abundances within these intensive study areas vary with site area rather than habitat quality. Constructing a revised habitat model based on vireo point count data may improve estimates of habitat quality and associated vireo density.

Desert Tortoise Case Study

Desert Tortoise Habitat Model

The habitat model generated by MaxEnt was relatively accurate, with an AUC of 0.81 on test data (Figure 44). The two most important variables for the model were vegetation and slope (Table 14). Evergreen subdesert shrubland and barren land were the two vegetation classes with the highest probability of desert tortoise presence. Probability of tortoise presence was highest on slopes between 5-20 degrees.

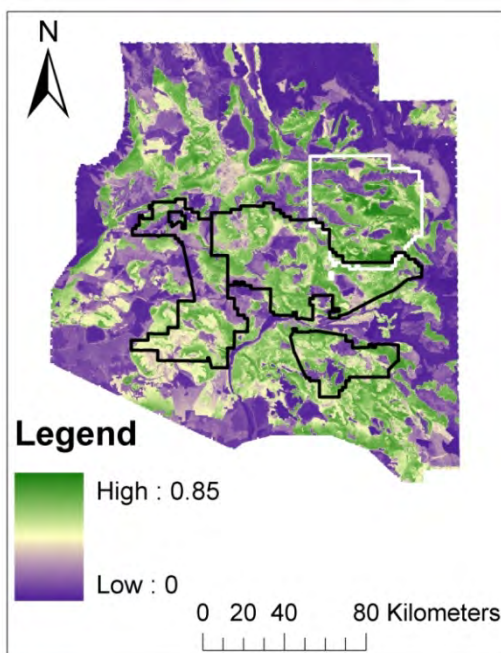


Figure 44. Habitat model for desert tortoise generated using MaxEnt. Presence data are from deserttortoise.gov and records maintained by Fort Irwin, CA. Green colors indicate high habitat probability, while purple colors indicate low habitat probability.

Table 14. Variable importance in the desert tortoise MaxEnt habitat model. Higher values indicate higher importance to the model.

Variable	Percent contribution
Vegetation	25.3
Slope	18.8
Soil 20cm	16.9
Soil 60cm	14.5
Soil 40cm	11.8
Aspect	6.4
Soil 80cm	3.5
Soil 10cm	1.6
Soil 100cm	0.8
Soil 30cm	0.2
Soil 5cm	0.1

These relationships agree with previous research suggesting that desert tortoises are often found in gently sloping areas with sparse cover of evergreen shrubs (U.S. Fish and Wildlife Service 2008). The high probability of desert tortoise habitat in barren areas likely reflects the

areas where abundant herbaceous plant life would be found, as herbaceous plants are the desert tortoises' primary food source.

Predictions from the habitat model overlap those of a recently published desert tortoise habitat model (Nussear et al. 2009). The habitat model in the present study covered a smaller extent and was sampled at a finer grain than the Nussear et al. (2009) model. However, agreement between the two models was relatively high. Both models used MaxEnt, but different data were used in each model, suggesting that both models were able to capture some important aspects of desert tortoise habitat requirements.

Simulating Population Responses to Upper Respiratory Tract Disease and Climate Change

Desert tortoise populations in the absence of disease and with a stable climate reached an average density of 5.50 tortoises per km² (Figure 45). Climate alone had no effect on average density of tortoises (5.50 tortoises per km², data not shown). The population trajectory of desert tortoises varied based on the assumption of survival rates when URTD was present in the population (Figure 45). For example, given a maximum survival rate of 93.75%, average tortoise density was 0.13 tortoises per km², whereas given a survival rate of 92% for adult tortoises, average density was 0.11 tortoises per km² (Figure 45). Assuming a synergistic interaction between rainfall and URTD resulted in average tortoise densities of approximately 0.05 tortoises per km², regardless of the emissions scenario, climate model, or survival assumptions (91-93.75% maximum survival).

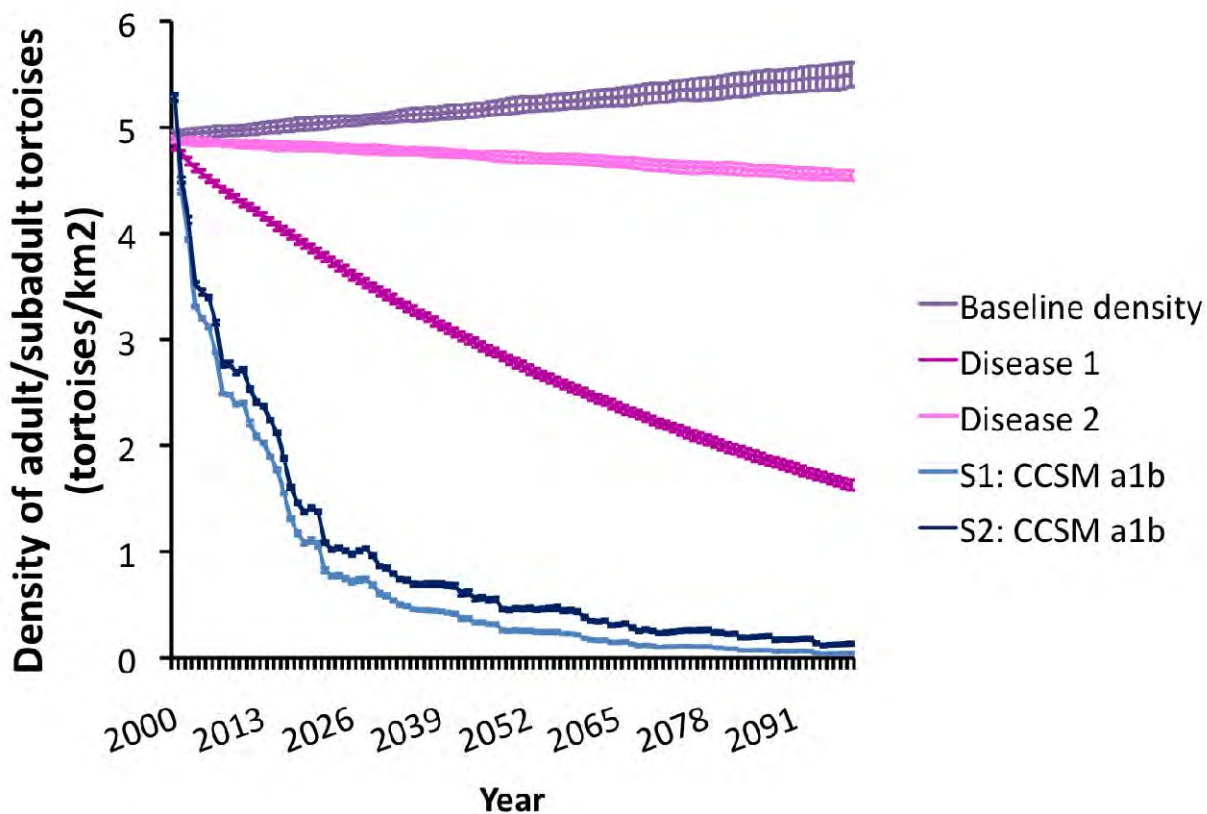


Figure 45. Effects of climate and URTD on desert tortoise density in the Western Mojave.

Desert tortoise survival was negatively impacted by the presence of URTD in simulated populations. Assuming an interaction between climate and URTD mortality resulted in extremely low tortoise densities by the end of the century. The rate of decline in the synergistic scenarios was approximately 50% in the first 15 years.

Although 50% population declines seem dramatic, survey data from 1979-1992 found rates of declines up to 90% (Berry 1997). Thus, the current results may underestimate the effects of URTD and precipitation. This assertion was supported by hind-casting tortoise densities from 1975-2000 using historic climate data, where rates of declines were approximately 50% over the modeled period as well. Therefore, the current simulation likely underestimates the potential effect of URTD and drought on desert tortoises in the Western Mojave.

The results also suggest that URTD alone is not likely to have caused the dramatic population declines observed over the species range. Survival of desert tortoises was lower when URTD was present in the populations, as seen by reduced densities of tortoises by the year 2100. However, the rate of decline in the URTD-alone scenarios was much slower than the observed declines (Berry 1997). It is possible that the parameterization of the model underestimated the effect of disease alone. More research into the effects of URTD on desert tortoises is necessary before a more accurate model can be built. Recent work on URTD in gopher tortoises should be used to guide work on the desert tortoises to help elucidate the effects of URTD on individual survival (e.g., Ozgul et al. 2009).

Interestingly, climate alone had no effect on desert tortoise density in the current simulations. Although precipitation was projected to change in the study area, the total amount of rainfall from September through March did not change dramatically, and consequently there was little effect of climate on the tortoise.

The current study results should be interpreted with caution. Very little is known about vital rates and other parameters in the model for the desert tortoises. The cryptic nature of desert tortoises has made population assessments extremely difficult. Tortoise populations have been surveyed using a number of different techniques, including large plot surveys, small plot surveys, and line-distance sampling; however, all of these techniques have drawbacks and may not provide reliable estimates of desert tortoise populations (Freilich et al. 2005). For example, line distance sampling was highly biased and may not be able to detect population declines of up to 50% (Freilich et al. 2005). Thus, estimates of tortoise density may be off by as much as 50%. In addition, almost nothing is known about the vital rates of neonatal and juvenile tortoises. The current study used the values in Doak et al. (1994). Population trajectory of desert tortoises is believed to depend primarily on survival of large adult females (Heppell 1998); however, other age classes (size classes) could have strong effects on population trends (Wisdom et al. 2000).

Despite the problems with unknown parameters within the current study, the relative effects of climate and disease reveal a potential interaction between URTD and precipitation. The results of the current model should be experimentally tested and field verified using careful plot design. In addition, management strategies to help tortoises cope with low water years should be designed and implemented. URTD is likely present in most desert tortoise populations at low levels even when the disease is not detected through ELISA or serology. Thus, mitigating the reduced body condition due to low precipitation and low food levels may help the desert tortoise persist.

Objective 3: Provide the DoD with the improved, highly flexible, user-friendly modeling tool, a comprehensive user's manual, and a training workshop on using the model.

The HexSim software is freely available and can be downloaded from one of two websites:

www.hexsim.net

www.epa.gov/hexsim

HexSim only runs on Microsoft's Windows operating system. However, the model engine is written in C++, so porting it to another platform is possible. The HexSim GUI is written in C#, and would have to be completely rewritten for non-Windows computers. The HexSim model engine has been compiled for both 32-bit and 64-bit computers, and HexSim.exe will automatically launch the appropriate version. HexSim does not use an installer -- you simply download the program and run it. This means it is not necessary to have administrative permissions on your computer to use HexSim.

The HexSim User's Guide is available in two formats: Web-Help and PDF. The two are almost interchangeable. The Web-Help document is the guide's native format, and the conversion to PDF introduced some awkward page breaks and other layout irregularities. The Web-Help document opens in a web browser, and it includes videos not available in the PDF version. User's Guides can always be improved, and this document is no exception. However, the present version is complete and highly detailed. At 304 pages (the PDF version), this document provides readers with very thorough coverage of all aspects of the HexSim model (Appendix B). The HexSim User's Guide is complete, but will be improved over time.

HexSim is also bundled with an additional PDF document titled "Getting Started", a collection of accessories, and an example workspace. The Getting Started document describes the accessories and examples, but the latter is described in detail in the User's Guide as well.

Conclusions and Implications for Future Research/Implementation

The HexSim Modeling Platform

The HexSim modeling platform can be applied to both simple and complex research questions to address management needs for a wide variety of wildlife species. The model can be used to investigate the potential impacts of different management, land-use, climate-change, or other scenarios. It can be used to identify critical habitat, plan for connectivity, assess population viability, and explore potential impacts of competitors, predators, pollutants, and invasive species. Fruitful future research could involve applying the model to address management relevant questions on different DoD installations.

HexSim is currently being applied to explore the drivers and the implications of source-sink population dynamics for at-risk species in changing landscapes (SERDP project RC-2120). For that project, it is being used both to explore and develop theory as well as to simulate existing populations of Black-capped Vireos at Fort Hood. HexSim has also been recently applied to model populations of Ord's kangaroo rat, Spotted Owls, and elk.

Although the model was recently demonstrated to members of the DoD Conservation Committee, there is likely a need to conduct additional technology transfer both within and outside of the military. Such transfer could take the form of workshops at scientific meetings or other venues.

The Red-cockaded Woodpecker at Fort Benning

The simulations of RCW populations at Fort Benning demonstrated the importance of forest management practices and future range development for meeting conservation goals. Additional research could better inform how best to continue longleaf pine restoration. SERDP has funded at least one project (SI-1303) exploring restoration approaches similar to the "interplant" scenario explored in the RCW population simulations.

Although the vegetation modeling and the RCW simulations showed little impact of projected climatic changes in RCWs, these efforts only explored two potential climate change impacts—climate-driven changes in habitat and the effects of spring precipitation regimes on reproduction. These modeling efforts did not consider the effects of potential changes in the insect community, the bird community, or the predator community. Nor did they consider changes in pine disease and pest dynamics. Further research into how climate change might affect RCWs at Fort Benning could provide a more thorough forecast of potential climate impacts.

Black-capped Vireos at Fort Hood

The Fort Hood Black-capped Vireo population is conservation-reliant (Scott et al. 2005). Cessation of cowbird trapping would result in a dramatic decline in the abundance of vireos that would likely not be compensated for by increased habitat availability. Reduced cattle stocking rates may provide an alternative to cowbird trapping, but a short-term study suggested that impacts would only be significant if stocking adjustments included both Fort Hood and surrounding rangelands (Kostecke et al. 2003). This may be politically infeasible.

When cowbirds are controlled, climate-induced increases in area burned have the potential to increase vireo habitat availability, whereas a policy of fire suppression could result in

long-term declines in vireo populations. Future climate-induced increases in area burned may be substantially larger than those modeled here, suggesting a potential risk for management of the Fort Hood Golden-cheeked Warbler population. Proposed increases in military training are likely to have minimal impact on long-term vireo abundances on Fort Hood. However, training may interact additively with climate-induced changes in fire to increase vireo abundances.

The Desert Tortoise at Fort Irwin

The results of the desert tortoise population simulations at Fort Irwin indicated that the population is likely to continue to decline—at a rate similar to that experienced over the last 20-30 years. The interaction between precipitation (which in turn affects forage quantity and quality) and the effects of URTD may provide some suggestions for potential management actions. It may be possible to offset the effects of the disease by providing high quality forage, but further study is needed.

Literature Cited

- Allouche, O., A. Tsoar, and R. Kadmon. 2006. Assessing the accuracy of species distribution models: prevalence, kappa and the true skill statistic (TSS). *Journal of Applied Ecology* **43**:1223-1232.
- Andersen, M. C., J. M. Watts, J. E. Freilich, S. R. Yool, G. I. Wakefield, J. F. McCauley, and P. B. Fahnestock. 2000. Regression-tree modeling of desert tortoise habitat in the central Mojave Desert. *Ecological Applications* **10**:890-900.
- Anderson, D. R., K. P. Burnham, B. C. Lubow, L. Thomas, P. S. Corn, P. A. Medica, and R. W. Marlow. 2001. Field trials of line transect methods applied to estimation of desert tortoise abundance. *Journal of Wildlife Management* **65**:583-597.
- Araújo, M. B. and M. New. 2007. Ensemble forecasting of species distributions. *Trends in Ecology and Evolution* **22**:42-47.
- Araújo, M. B., R. G. Pearson, W. Thuiller, and M. Erhard. 2005. Validation of species–climate impact models under climate change. *Global Change Biology* **11**:1504-1513.
- Bachelet, D., J. Lenihan, R. Drapek, and R. Neilson. 2008. VEMAP vs VINCERA: A DGVM sensitivity to differences in climate scenarios. *Global and Planetary Change* **64**:38-48.
- Bailey, J. W. and F. R. Thompson. 2007. Multiscale nest-site selection by black-capped vireos. *Journal of Wildlife Management* **71**:828-836.
- Bancroft, B., C. Wilsey, and J. Lawler. *Submitted*. A test of single and multi-scale ensemble approaches for predicting habitat suitability. *Ecography*.
- Barber, D. R. and T. E. Martin. 1997. Influence of alternate host densities on brown-headed cowbird parasitism rates in black-capped vireos. *The Condor* **99**:595-604.
- Battin, J. and J. J. Lawler. 2006. Cross-scale correlations and the design and analysis of avian habitat selection studies. *Condor* **108**:59-70.
- Beatley, J. C. 1974. Phenological events and their environmental triggers in Mojave Desert ecosystems. *Ecology* **55**:856-863.
- Beebee, T. J. C. 1995. Amphibian breeding and climate. *Nature* **374**:219-220.
- Berry, K. H. 1997. Demographic consequences of disease in two desert tortoise populations in California, USA. Pages 91-99 *in* Conservation, Restoration, and Management of Tortoises and Turtles--An International Conference. New York Turtle and Tortoise Society.
- Berry, K. H., T. Y. Bailey, and K. M. Anderson. 2006. Attributes of desert tortoise populations at the National Training Center, Central Mojave Desert, California, USA. *Journal of Arid Environments* **67**:165-191.
- Bian, L. and R. Butler. 1999. Comparing effects of aggregation methods on statistical and spatial properties of simulated spatial data. *Photogrammetric Engineering and Remote Sensing* **65**:73-84.
- Boarman, W. I., M. A. Patten, R. J. Camp, and S. J. Collis. 2006. Ecology of a population of subsidized predators: Common ravens in the central Mojave Desert, California. *Journal of Arid Environments* **67 Supplement**:248-261.
- Breiman, L. 2001. Random forests. *Machine Learning* **45**:5-32.
- Burnham, K. P. and D. R. Anderson. 2002. Model selection and multimodel inference : a practical information-theoretic approach. Springer, New York.
- Carey, V. 2007. ROC: utilities for ROC, with uarray focus v1.16.0. Bioconductor, <http://www.bioconductor.org>.

- Chambers, J. M. and T. J. Hastie. 1991. Statistical Models in S. Chapman and Hall, Boca Raton, Florida.
- Christopher, M. M., K. A. Nagy, I. R. Wallis, J. K. Klaassen, and K. H. Berry. 1997. Laboratory health profiles of desert tortoises in the Mojave Desert: a model for health status evaluation of chelonian populations. Pages 76-82 *in* Conservation, Restoration, and Management of Tortoises and Turtles--An International Conference. New York Turtle and Tortoise Society.
- Cimprich, D. A. 2009. Monitoring of the black-capped vireo during 2009 on Fort Hood, Texas. Endangered species monitoring and management at Fort Hood, Texas: 2009 annual report. The Nature Conservancy, Fort Hood Project, Fort Hood, Texas, USA.
- Cimprich, D. A. and K. Comolli. 2010. Monitoring of the black-capped vireo during 2010 on Fort Hood, Texas. Endangered species monitoring and management at Fort Hood, Texas: 2010 annual report. The Nature Conservancy, Fort Hood Project, Fort Hood, Texas, USA.
- Cimprich, D. A. and R. M. Kostecke. 2006. Distribution of the black-capped vireo at Fort Hood, Texas. *The Southwestern Naturalist* **51**:99-102.
- Collins, W. D., C. M. Bitz, M. L. Blackmon, G. B. Bonan, C. S. Bretherton, J. A. Carton, P. Chang, S. C. Doney, J. J. Hack, T. B. Henderson, J. T. Kiehl, W. G. Large, D. S. McKenna, B. D. Santer, and R. D. Smith. 2006. The community climate system model version 3 (CCSM3). *Journal of Climate* **19**:2122-2143.
- Confalonieri, R., G. Bellocchi, S. Bregaglio, M. Donatelli, and M. Acutis. 2010. Comparison of sensitivity analysis techniques: A case study with the rice model WARM. *Ecological Modelling* **221**:1897-1906.
- Conner, R. N., D. C. Rudolph, R. R. Schaefer, D. Saenz, and C. E. Shackelford. 1999. Relationships among red-cockaded woodpecker group density, nestling provisioning rates, and habitat. *Wilson Bulletin* **111**:494-498.
- Conner, R. N., D. C. Rudolph, and J. R. Walters. 2001. Social behavior and population biology. Pages 117-171 *The red-cockaded woodpecker: surviving in a fire-maintained ecosystem*. University of Texas Press, Austin.
- Conner, R. N., D. Saenz, R. R. Schaefer, J. R. McCormick, D. C. Rudolph, and D. B. Burt. 2004. Group size and nest success in Red-cockaded woodpeckers in the West Gulf Coastal Plain: helpers make a difference. *Journal of Field Ornithology* **75**:74-78.
- Conner, R. N., D. Saenz, R. R. Schaefer, J. R. McCormick, D. C. Rudolph, and D. B. Burt. 2005. Rainfall, El Niño, and reproduction of Red-cockaded woodpeckers. *Southeastern Naturalist* **4**:347-354.
- Crick, H. Q. P. and T. H. Sparks. 1999. Climate change related to egg-laying trends. *Nature* **399**:423-423.
- Cutler, D. R., T. C. Edwards Jr., K. H. Beard, A. Cutler, K. T. Hess, J. Gibson, and J. J. Lawler. 2007. Random forests for classification in ecology. *Ecology* **88**:2783-2792.
- Damgaard, C., J. Weiner, and H. Nagashima. 2002. Modelling individual growth and competition in plant populations: growth curves of *Chenopodium album* at two densities. *Journal of Ecology* **90**:666-671.
- Daniels, S. J. and J. R. Walters. 2000. Between-year breeding dispersal in red-cockaded woodpeckers: multiple causes and estimated cost. *Ecology* **81**:2473-2484.
- DeBoer, T. S. and D. D. Diamond. 2006. Predicting presence-absence of the endangered golden-cheeked warbler (*Dendroica chrysoparia*). *The Southwestern Naturalist* **51**:181-190.

- DeCapita, M. E. 2000. Brown-headed cowbird control on Kirtland's warbler nesting areas in Michigan, 1972-1995. Pages 333-341 in J. N. M. Smith, T. L. Cook, S. I. Rothstein, S. K. Robinson, and S. G. Sealy, editors. Ecology and management of cowbirds and their hosts: studies in the conservation of North American passerine birds. University of Texas Press, Austin, TX.
- DeLotelle, R. S. and R. J. Epting. 1992. Reproduction of the red-cockaded woodpecker in central Florida. *Wilson Bulletin* **104**:285-294.
- Doak, D., P. Kareiva, and B. Klepetka. 1994. Modeling population viability for the desert tortoise in the western Mojave Desert. *Ecological Applications* **4**:446-460.
- Dolbeer, R. A. 1982. Migration patterns for age and sex classes of blackbirds and starlings. *Journal of Field Ornithology* **53**:28-46.
- Doresky, J., K. Morgan, L. Ragsdale, and H. Townsend. 2001. Effects of military activity on reproductive success of Red-cockaded Woodpeckers. *Journal of Field Ornithology* **72**:305-311.
- Doresky, J. K., M. G. Barron, and P. K. Swiderek. 2003. Landscape scale restoration and red-cockaded woodpecker recovery? Pages 127-133 in R. Costa and S. J. Daniels, editors. Red-cockaded woodpecker: road to recovery. Hancock House Publishers, Blaine, WA.
- Duke, C. S., C. Mauldin, and H. Balbach. 2008. Forest health in the Southeastern United States: Assessment of the state of the science. ERDC/CERL SR-08-9.
- Eckhardt, L. G., M. A. S. Sayer, and D. Imm. 2010. State of pine decline in the Southeastern United States. *Southern Journal of Applied Forestry* **34**:138-141.
- Eckrich, G. H., J. A. Koloszar, and M. D. Goering. 1999. Effective landscape management of brown-headed cowbirds at Fort Hood, Texas. *Studies in Avian Biology* **18**:267-274.
- Elith, J., C. H. Graham, R. P. Anderson, M. Dudik, S. Ferrier, A. Guisan, R. J. Hijmans, F. Huettmann, J. R. Leathwick, A. Lehmann, J. Li, L. G. Lohmann, B. A. Loiselle, G. Manion, C. Moritz, M. Nakamura, Y. Nakazawa, J. Overton, A. Townsend Peterson, S. J. Phillips, K. Richardson, R. Scachetti-Pereira, R. E. Schapire, J. Soberon, S. Williams, M. S. Wisz, and N. E. Zimmermann. 2006. Novel methods improve prediction of species' distributions from occurrence data. *Ecography* **29**:129-151.
- Engstrom, R. T. and F. J. Sanders. 1997. Red-cockaded woodpecker foraging ecology in an old-growth longleaf pine forest. *Wilson Bulletin* **109**:203-217.
- ESRI. 2008. ArcMap 9.3, Redlands, CA, USA.
- ESRI. 2009. ArcGIS 9.3.1. ESRI, Redlands, CA.
- Fielding, A. H. and J. F. Bell. 1997. A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environmental Conservation* **24**:38-49.
- Final Biological Assessment for Proposed Maneuver Center of Excellence Actions at Fort Benning Georgia. 2008. Prepared by U.S. Army Corps of Engineers. 27 October 2008. Available as "MCOE BA Final 11-5-08" at http://www.hqda.army.mil/acsim/brac/nepa_EIS_docs.htm.
- Fitzgerald, R. W. and B. G. Lees. 1994. Assessing the classification accuracy of multisource remote sensing data. *Remote Sensing of Environment* **47**:362-368.
- Flannigan, M. D., K. A. Logan, B. D. Amiro, W. R. Skinner, and B. J. Stocks. 2005. Future area burned in Canada. *Climatic Change* **72**:1-16.
- Freeman, E. A. and G. Moisen. 2008a. A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa. *Ecological Modelling* **217**:48-58.

- Freeman, E. A. and G. Moisen. 2008b. PresenceAbsence: an R package for presence absence analysis. *Journal of Statistical Software* **23**:31.
- Freilich, J. E., R. J. Camp, and J. J. Duda. 2005. Problems with sampling desert tortoises: a simulation analysis based on field data. *Journal of Wildlife Management* **69**:45-56.
- Fuhlendorf, S. D., F. E. Smeins, and W. E. Grant. 1996. Simulation of a fire-sensitive ecological threshold: A case study of Ashe juniper on the Edwards plateau of Texas, USA. *Ecological Modelling* **90**:245-255.
- Garzón, M. B., R. Blazek, M. Neteler, R. Sánchez de Dios, H. S. Ollero, and C. Furlanello. 2006. Predicting habitat suitability with machine learning models: The potential area of *Pinus sylvestris* L. in the Iberian Peninsula. *Ecological Modelling* **197**:383-393.
- Gelbard, J. L. and J. Belnap. 2003. Roads as conduits for exotic plant invasions in a semiarid landscape. *Conservation Biology* **17**:420-432.
- Goering, M. D. 1998. Mapping the extent of the 1996 wildfire on Fort Hood using LANDSAT TM satellite imagery, color infrared photography and GIS. Summary of 1997 Research Activities, Texas Conservation Data Center. The Nature Conservancy, Fort Hood Project, Fort Hood, Texas, USA.
- Goguen, C. B. and N. E. Mathews. 2000. Local gradients of cowbird abundance and parasitism relative to livestock grazing in a western landscape. *Conservation Biology* **14**:1862-1869.
- Goguen, C. B. and N. E. Mathews. 2001. Brown-headed cowbird behavior and movements in relation to livestock grazing. *Ecological Applications* **11**:1533-1544.
- Graber, J. W. 1961. Distribution, habitat requirements, and life history of the black-capped vireo (*Vireo atricapilla*). *Ecological Monographs* **31**:313-336.
- Graf, R. F., K. Bollmann, W. Suter, and H. Bugmann. 2005. The importance of spatial scale in habitat models: capercaillie in the Swiss Alps. *Landscape Ecology* **20**:703-717.
- Graf, R. F., L. Mathys, and K. Bollmann. 2009. Habitat assessment for forest dwelling species using LiDAR remote sensing: Capercaillie in the Alps. *Forest Ecology and Management* **257**:160-167.
- Grimm, V. and S. F. Railsback. 2005. Individual-based modeling and ecology. Princeton University Press, Princeton.
- Grimm, V., E. Revilla, U. Berger, F. Jeltsch, W. M. Mooij, S. F. Railsback, H. H. Thulke, J. Weiner, T. Wiegand, and D. L. DeAngelis. 2005. Pattern-oriented modeling of agent-based complex systems: Lessons from ecology. *Science* **310**:987-991.
- Grzybowski, J. A., D. J. Tazik, and G. D. Schnell. 1994. Regional analysis of black-capped vireo breeding habitats. *The Condor* **96**:512-544.
- Hall, L. S. and S. I. Rothstein. 1999. Cowbird control: the efficacy of long-term control and proposed alternatives to standard control practices. *Studies in Avian Biology* **18**:254-259.
- Hanley, J. A. and B. J. McNeil. 1982. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **143**:29-36.
- Hanula, J. L., K. E. Franzreb, and W. D. Pepper. 2000. Longleaf pine characteristics associated with arthropods available for red-cockaded woodpeckers. *Journal of Wildlife Management* **64**:60-70.
- Hastie, T. J. and R. J. Tibshirani. 1990. Generalized Additive Models. Chapman and Hall, London.
- Henen, B. T., C. C. Peterson, I. R. Wallis, K. H. Berry, and K. A. Nagy. 1998. Effects of climatic variation on field metabolism and water relations of desert tortoises. *Oecologia* **117**:365-373.

- Heppell, S. S. 1998. Application of life-history theory and population model analysis to turtle conservation. *Copeia* **1998**:367-375.
- Hilborn, R. and M. Mangel. 1997. *The Ecological Detective*. Princeton University Press, Princeton.
- Hooper, R. G. 1996. Arthropod biomass in winter and the age of longleaf pines. *Forest Ecology and Management* **82**:115-131.
- Hothorn, T., K. Hornik, C. Strobl, and A. Zeileis. 2010. Package 'party'.
- Houghton, J. T., Y. Ding, D. J. Griggs, M. Noguer, P. J. van der Linden, X. Dai, K. Maskell, and C. A. Johnson, editors. 2001. *Climate Change 2001: The Scientific Basis*. Cambridge University Press, Cambridge.
- Hovis, J. A. and R. F. Labisky. 1985. Vegetative associations of red-cockaded woodpecker colonies in Florida. *Wildlife Society Bulletin* **13**:307-314.
- Jackson, N. H. and D. D. Roby. 1992. Fecundity and Egg-Laying Patterns of Captive Yearling Brown-Headed Cowbirds. *The Condor* **94**:585-589.
- Jiménez-Valverde, A. and J. M. Lobo. 2007. Threshold criteria for conversion of probability of species presence to either-or presence-absence. *Acta Oecologia* **31**:361-369.
- Johnson, F. L. 1982. Effects of Tank Training Activities on Botanical Features at Fort Hood, Texas. *The Southwestern Naturalist* **27**:309-314.
- Keane, R. E., M. Rollins, and Z. L. Zhu. 2007. Using simulated historical time series to prioritize fuel treatments on landscapes across the United States: The LANDFIRE prototype project. *Ecological Modelling* **204**:485-502.
- Kelly, J. F., S. M. Pletschet, and D. M. Leslie Jr. 1993. Habitat associations of red-cockaded woodpecker cavity trees in an old-growth forest of Oklahoma. *Journal of Wildlife Management* **57**:122-128.
- Komuro, R., E. D. Ford, and J. H. Reynolds. 2006. The use of multi-criteria assessment in developing a process model. *Ecological Modelling* **197**:320-330.
- Kostecke, R. M. 2008. Population Trends of Breeding Birds on the Edwards Plateau, Texas: Local Versus Regional Patterns. *Southwestern Naturalist* **53**:466-471.
- Kostecke, R. M. and D. A. Cimprich. 2008. Adult and juvenile survival of black-capped vireos within a large breeding population in Texas. *The Condor* **110**:251-259.
- Kostecke, R. M., J. A. Koloszar, and D. C. Dearborn. 2003. Effect of a reduction in cattle stocking rate on brown-headed cowbird activity. *Wildlife Society Bulletin* **31**:1083-1091.
- Kostecke, R. M., S. G. Summers, G. H. Eckrich, and D. A. Cimprich. 2005. Effects of Brown-headed Cowbird (*Molothrus ater*) removal on Black-capped Vireo (*Vireo atricapilla*) nest success and population growth at Fort Hood, Texas. *Ornithological Monographs* **57**:28-37.
- Küchler, A. W. 1993. Potential natural vegetation of the conterminous United States. Global Ecosystems Database Version 2.0. Digital vector data in an Albers Equal Area Conic polygon network and derived raster data on a 5 km by 5 km Albers equal area 590 · 940 Grid, one independent and one dependent single-attribute spatial layer, 3,580,905 bytes in 13 files. National Oceanic and Atmospheric Administration, National Geophysical Data Center, Boulder, Colorado, USA.
- Kus, B. E. 1999. Impacts of brown-headed cowbird parasitism on productivity of the endangered Least Bell's Vireo. *Studies in Avian Biology* **18**:160-166.
- LaBranche, M. S. and J. R. Walters. 1994. Patterns of mortality in nests of red-cockaded woodpeckers in the sandhills of southcentral North Carolina. *Wilson Bulletin* **106**:258-271.

- Landis, J. R. and G. G. Koch. 1977. Measurement of Observer Agreement for Categorical Data. *Biometrics* **33**:159-174.
- Lawler, J. J., S. P. Campbell, A. D. Guerry, M. B. Koloszy, R. J. O'Connor, and L. C. N. Seward. 2002. The scope and treatment of threats in endangered species recovery plans. *Ecological Applications* **12**:663-667.
- Lawler, J. J., D. White, R. P. Neilson, and A. R. Blaustein. 2006a. Predicting climate-induced range shifts: model differences and model reliability. *Global Change Biology* **12**:1568-1584.
- Lawler, J. L., D. White, R. P. Neilson, and A. R. Blaustein. 2006b. Predicting climate-induced range shifts: model differences and model reliability. *Global Change Biology* **12**:1568-1584.
- Lederle, P. E., K. R. Rautenstrauch, D. L. Rakestraw, K. K. Zander, and J. L. Boone. 1997. Upper respiratory tract disease and mycoplasmosis in desert tortoises from Nevada. *Journal of Wildlife Diseases* **33**:759-765.
- Lenihan, J., D. Bachelet, R. Neilson, and R. Drapek. 2008. Response of vegetation distribution, ecosystem productivity, and fire to climate change scenarios for California. *Climatic Change* **87**:S215-S230.
- Leonard, D. L., R. S. DeLotelle, and R. J. Epting. 2003. Factors contributing to variation in fledgling production in central Florida red-cockaded woodpeckers. Pages 593-602 in R. Costa and S. J. Daniels, editors. *Red-cockaded Woodpecker: Road to Recovery*. Hancock House Publishers, Blaine, WA.
- Liaw, A. and M. Wiener. 2002. Classification and Regression by randomForest. *R News* **2**:18-22.
- Littell, J. S., D. McKenzie, D. L. Peterson, and A. L. Westerling. 2009. Climate and wildfire area burned in western U. S. ecoprovinces, 1916-2003. *Ecological Applications* **19**:1003-1021.
- Lobo, J. M., A. Jimenez-Valverde, and R. Real. 2008. AUC: a misleading measure of the performance of predictive distribution models. *Global Ecology and Biogeography* **17**:145-151.
- Longshore, K. M., J. R. Jaeger, and J. M. Sappington. 2003. Desert tortoise (*Gopherus agassizii*) survival at two Eastern Mojave Desert sites: death by short-term drought? *Journal of Herpetology* **37**:169-177.
- Mack, M. C. and C. M. D'Antonio. 1998. Impacts of biological invasions on disturbance regimes. *Trends in Ecology and Evolution* **13**:195-198.
- Manel, S., H. C. Williams, and S. J. Ormerod. 2001. Evaluating presence-absence models in ecology: the need to account for prevalence. *Journal of Applied Ecology* **38**:921-931.
- Marmion, M., M. Parviainen, M. Luoto, R. K. Heikkinen, and W. Thuiller. 2009. Evaluation of consensus methods in predictive species distribution modelling. *Diversity and Distributions* **15**:59-69.
- McComb, B., T. A. Spies, and K. A. Olsen. 2007. Sustaining biodiversity in the Oregon Coast Range: potential effects of forest policies in a multi-ownership province. *Ecology and Society* **12**:29-49.
- McGaughey, R. J. 2009. FUSION/LDV: Software for LIDAR Data Analysis and Visualization. US Department of Agriculture Forest Service, Pacific Northwest Research Station, University of Washington, Seattle, WA.
- McKenzie, D., Z. Gedalof, D. L. Peterson, and P. Mote. 2004a. Climatic change, wildfire, and conservation. *Conservation Biology* **18**:890-902.

- Meehl, G. A., C. Covey, T. Delworth, M. Latif, B. McAvaney, J. F. B. Mitchell, R. J. Souffer, and K. E. Taylor. 2007. The WCRP CMIP3 multimodel dataset: A new era in climate change research. *Bulletin of the American Meteorological Society* **88**:1383-1394.
- Meehl, G. A., C. Covey, B. McAvaney, M. Latif, and R. J. Stoufer. 2005. Overview of the coupled model intercomparison project. *Bulletin of the American Meteorological Society* **86**:89-93.
- Meyer, D., A. Seileis, and K. Hornik. 2010. vcd: Visualizing Categorical Data 1.2-8. CRAN <http://cran.r-project.org/web/packages/>.
- Miller, D. A. and R. A. White. 1998. A conterminous United States multi-layer soil characteristics data set for regional climate and hydrology modeling. *Earth Interactions* **2**: [Available on-line at <http://earthinteractions.org>].
- Milly, P. C. D., K. A. Dunne, and A. V. Vecchia. 2005. Global pattern of trends in streamflow and water availability in a changing climate. *Nature* **438**:347-350.
- Mitchell, T. D. and P. D. Jones. 2005. An improved method of constructing a database of monthly climate observations and associated high-resolution grids. *International Journal of Climatology* **25**:693-712.
- Mullen, E. B. and P. Ross. 1997. Survival of relocated tortoises: feasibility of relocating tortoises as a successful mitigation tool. Pages 140-146 in *Conservation, Restoration, and Management of Tortoises and Turtles--An International Conference*. New York Turtle and Tortoise Society.
- Nakicenovic, N., J. Alcamo, G. Davis, B. d. Vries, J. Fenhann, S. Gaffin, K. Gregory, A. Grübler, T. Y. Jung, T. Kram, E. L. L. Rovere, L. Michaelis, S. Mori, T. Morita, W. Pepper, H. Pitcher, L. Price, K. Riahi, A. Roehrl, H.-H. Rogner, A. Sankovski, M. Schlesinger, P. Shukla, S. Smith, R. Swart, S. v. Rooijen, N. Victor, and Z. Dadi. 2000. *Special Report on Emissions Scenarios. A Special Report of Working Group III of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, UK.
- Neal, J. C., D. A. James, W. G. Montague, and J. E. Johnson. 1993. Effects of weather and helpers on survival of nestling red-cockaded woodpeckers. *Wilson Bulletin* **105**:666-673.
- New, M., M. Hulme, and P. D. Jones. 1999. Representing twentieth century space-time climate variability. Part 1: development of a 1961-90 mean monthly terrestrial climatology. *Journal of Climate* **12**:829-856.
- New, M., D. Lister, M. Hulme, and I. Makin. 2002. A high-resolution data set of surface climate over global land areas. *Climate Research* **21**:1-25.
- Noble, M. E., W. S. Webb, S. G. Summers, and E. W. Runfeldt. 2008. *Habitat management activities, Fort Hood, Texas, 2008. Endangered species monitoring and management at Fort Hood, Texas: 2003 annual report*. The Nature Conservancy, Fort Hood Project, Fort Hood, Texas, USA.
- Nussear, K. E., T. C. Esque, R. D. Inman, L. Gass, K. A. Thomas, C. S. A. Wallace, J. B. Blainey, D. M. Miller, R. H. Webb, and U. S. G. S. O.-F. R.-. 2009, 18 p. 2009. Modeling habitat of the desert tortoise (*Gopherus agassizii*) in the Mojave and parts of the Sonoran Deserts of California, Nevada, Utah, and Arizona: U.S. Geological Survey Open-File Report 2009-1102.
- Optimal Geomatics. 2009. Fort Hood LiDAR Survey. Optimal Geomatics, Inc., Huntsville, AL.
- Ortega, J. C. and C. P. Ortega. 2009. Sex ratios and survival probabilities of Brown-headed Cowbirds (*Molothrus ater*) in southwest Colorado. *Auk* **126**:268-277.

- Ozgul, A., M. K. Oli, B. M. Bolker, and C. Perez-Heydrich. 2009. Upper respiratory tract disease, force of infection, and effects on survival of gopher tortoises. *Ecological Applications* **19**:786-798.
- Parendes, L. A. and J. A. Jones. 2000. Role of light availability and dispersal in exotic plant invasion along roads and streams in the H. J. Andrews Experimental Forest, Oregon. *Conservation Biology* **14**:64-75.
- Parmesan, C. 2006. Ecological and evolutionary responses to recent climate change. *Annual Review of Ecology Evolution and Systematics* **37**:637-669.
- Parmesan, C., N. Ryrholm, C. Stefanescu, J. K. Hill, C. D. Thomas, H. Descimon, B. Huntley, L. Kaila, J. Kullberg, T. Tammaru, W. J. Tennent, J. A. Thomas, and M. Warren. 1999. Poleward shifts in geographical ranges of butterfly species associated with regional warming. *Nature* **399**:579-583.
- Parmesan, C. and G. Yohe. 2003. A globally coherent fingerprint of climate change impacts across natural systems. *Nature* **421**:37-42.
- Payne, R. B. 1976. The Clutch Size and Numbers of Eggs of Brown-Headed Cowbirds: Effects of Latitude and Breeding Season. *The Condor* **78**:337-342.
- Pearson, R. G., W. Thuiller, M. B. Araújo, E. Martinez-Meyer, L. Brotons, C. McClean, L. Miles, P. Segurado, T. P. Dawson, and D. C. Lees. 2006. Model-based uncertainty in species range prediction. *Journal of Biogeography* **33**:1704-1711.
- Pekins, C. E. 2006. Armored military training and endangered species restrictions at Fort Hood, Texas. *Federal Facilities Environmental Journal* **2**:37-50.
- Peterson, C. C. 1994. Different rates and causes of high mortality in two populations of the threatened desert tortoise *Gopherus agassizii*. *Biological Conservation* **70**:101-108.
- Peterson, C. C. 1996. Ecological energetics of the desert tortoise (*Gopherus agassizii*): effects of rainfall and drought. *Ecology* **77**:1831-1844.
- Phillips, S. J., R. P. Anderson, and R. E. Schapire. 2006. Maximum entropy modeling of species geographic distributions. *Ecological Modelling* **190**:231-259.
- Pope, V. D., M. L. Gallani, P. R. Rowntree, and R. A. Stratton. 2000. The impact of new physical parameterizations in the Hadley Centre climate model -- HadAM3. *Climate Dynamics* **16**:123-146.
- Pounds, A. J., M. R. Bustamante, L. A. Coloma, J. A. Consuegra, M. P. L. Fogden, P. N. Foster, E. La Marca, K. L. Masters, A. Merino-Viteri, R. Puschendorf, S. R. Ron, G. A. Sanchez-Azofeifa, C. J. Still, and B. E. Young. 2006. Widespread amphibian extinctions from epidemic disease driven by global warming. *Nature* **439**:161-167.
- Pounds, J. A., M. P. L. Fogden, and J. H. Campbell. 1999. Biological response to climate change on a tropical mountain. *Nature* **398**:611-615.
- Prasad, A. M., L. R. Iverson, and A. Liaw. 2006. Newer classification and regression tree techniques: Bagging and random forests for ecological prediction. *Ecosystems* **9**:181-199.
- Prentice, I. C., G. D. Farquhar, M. J. R. Fasham, M. L. Goulden, M. Heimann, V. J. Jaramillo, H. S. Kheshgi, C. L. Quéré, R. J. Scholes, and D. W. R. Wallace. 2001. The carbon cycle and atmospheric carbon dioxide. Page 881 in J. T. Houghton, Y. Ding, D. J. Griggs, M. Noguer, P. J. v. d. Linden, X. Dai, K. Maskell, and C. A. Johnson, editors. *Climate Change 2001: The Scientific Basis. Contribution of Working Group I to the Third Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, United Kingdom.

- R Development Core Team. 2008. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL <http://www.R-project.org>.
- Reemts, C. M. and L. L. Hansen. 2008. Slow recolonization of burned oak-juniper woodlands by Ashe juniper (*Juniperus ashei*): Ten years of succession after crown fire. *Forest Ecology and Management* **255**:1057-1066.
- Reemts, C. M. and J. Teague. 2007. Vegetation classification and mapping of Fort Hood. Endangered species monitoring and management at Fort Hood, Texas: 2007 annual report. The Nature Conservancy, Fort Hood Project, Fort Hood, Texas, USA.
- Robinson, S. K. 1999. Cowbird ecology: factors affecting the abundance and distribution of cowbirds. *Studies in Avian Biology* **18**:4-9.
- Root, T. L., J. T. Price, K. R. Hall, S. H. Schneider, C. Rosenzweig, and J. A. Pounds. 2003. Fingerprints of global warming on wild animals and plants. *Nature* **421**:57-60.
- Rudolph, D. C. and R. N. Conner. 1991. Cavity tree selection by red-cockaded woodpeckers in relation to tree age. *Wilson Bulletin* **103**:458-467.
- Saltelli, A. 2004. Sensitivity analysis in practice : a guide to assessing scientific models. Wiley, Hoboken, NJ.
- Santos, M. J., J. A. Greenberg, and S. L. Ustin. 2010. Using hyperspectral remote sensing to detect and quantify southeastern pine senescence effects in red-cockaded woodpecker (*Picoides borealis*) habitat. *Remote Sensing of Environment* **114**:1242-1250.
- Schneider, S. H. and T. L. Root. 2002. *Wildlife Responses to Climate Change, North American Case Studies*. Island Press, Washington, D.C.
- Schumacher, I. M., D. B. Hardenbrook, M. B. Brown, E. R. Jacobson, and P. A. Klein. 1997. Relationship between clinical signs of upper respiratory tract disease and antibodies to *Mycoplasma agassizii* in desert tortoises from Nevada. *Journal of Wildlife Diseases* **33**:261-266.
- Schumaker, N. H. 1998. A user's guide to the PATCH model. EPA/600/R-98/135, U.S. Environmental Protection Agency, Corvallis, OR.
- Scinocca, J. F., N. A. McFarlane, M. Lazare, J. Li, and D. Plummer. 2008. The CCCma third generation AGCM and its extension into the middle atmosphere. *Atmospheric Chemistry and Physics* **8**:7055-7074.
- Scott, D. M. and C. D. Ankney. 1983. The laying cycle of brown-headed cowbirds: passerine chickens? *The Auk* **100**:583-592.
- Scott, J. M., D. D. Goble, J. A. Wiens, D. S. Wilcove, M. Bean, and T. Male. 2005. Recovery of imperiled species under the Endangered Species Act: the need for a new approach. *Frontiers in Ecology and the Environment* **3**:383-389.
- Seavy, N. E., J. H. Viers, and J. K. Wood. 2009. Riparian bird response to vegetation structure: a multiscale analysis using LiDAR measurements of canopy height. *Ecological Applications* **19**:1848-1857.
- Shafer, S. L., J. Atkins, B. A. Bancroft, P. J. Bartlein, J. J. Lawler, B. Smith, and C. B. Wilsey. In-Press. Projected Climate and Vegetation Changes and Potential Biotic Effects for Fort Benning, Georgia; Fort Hood, Texas; and Fort Irwin, California. Scientific Investigations Report, US Geological Survey 2011-5099.
- Sitch, S., B. Smith, I. C. Prentice, A. Arneth, A. Bondeau, W. Cramer, J. O. Kaplan, S. Levis, W. Lucht, M. T. Sykes, K. Thonicke, and S. Venevsky. 2003. Evaluation of ecosystem dynamics, plant geography and terrestrial carbon cycling in the LPJ dynamic global vegetation model. *Global Change Biology* **9**:161-185.

- Smith, B., I. C. Prentice, and M. T. Sykes. 2001. Representation of vegetation dynamics in the modelling of terrestrial ecosystems: comparing two contrasting approaches within European climate space. *Global Ecology and Biogeography* **10**:621-637.
- Smith, J. N. M. 1999. The basis for cowbird management: host selection, impact on hosts, and criteria for taking management action. *Studies in Avian Biology* **18**:104-108.
- Stake, M. M. and D. A. Cimprich. 2003. Using video to monitor predation at Black-capped Vireo nests. *Condor* **105**:348-357.
- Stehman, S. V. 1997. Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment* **62**:77-89.
- Strobl, C., A. L. Boulesteix, A. Zeileis, and T. Hothorn. 2007. Bias in random forest variable importance measures: Illustrations, sources and a solution. *Bmc Bioinformatics* **8**:.
- Swets, J. A. 1988. Measuring the Accuracy of Diagnostic Systems. *Science* **240**:1285-1293.
- Tazik, D. J., J. D. Cornelius, and C. A. Abrahamson. 1993. Status of the black-capped vireo at Fort Hood, Texas. US Army Construction and Engineering Research Laboratories, Champaign, IL.
- Thomas, C. D. and J. J. Lennon. 1999. Birds extend their ranges northwards. *Nature* **399**:213.
- Thuiller, W. 2004. Patterns and uncertainties of species' range shifts under climate change. *Global Change Biology* **10**:2020-2027.
- Thuiller, W., M. B. Araújo, and S. Lavorel. 2003. Generalized models vs. classification tree analysis: predicting spatial distributions of plant species at different scales. *Journal of Vegetation Science* **14**:669-680.
- Turner, F. B., P. A. Medica, and C. L. Lyons. 1984. Reproduction and survival of the desert tortoise (*Scaptochelys agassizii*) in Ivanpah Valley, California. *Copeia* **1984**:811-820.
- Tweddale, S. A. and T. D. Frank. 2006. A procedure to extrapolate vegetation cover estimates over large arid and semi-arid regions using multiple spatial resolution imagery. *Geocarto International* **21**:1-6.
- U.S. Fish and Wildlife Service. 2008. Draft revised recovery plan for the Mojave population of the desert tortoise (*Gopherus agassizii*). U.S. Fish and Wildlife Service, California and Nevada Region, Sacramento, California.
- U.S. Fish and Wildlife Service, R. 2003. Recovery plan for the red-cockaded woodpecker (*Picoides borealis*): second revision. U.S. Fish and Wildlife Service, Atlanta, GA.
- USDA-NRCS. 2007. US Department of Agriculture National Resources Conservation Service Soil Survey Geographic (SSURGO) Database. US Department of Agriculture.
- USDA-NRCS. 2008. US Department of Agriculture National Resources Conservation Service Soil Soil Data Viewer 5.2. US Department of Agriculture.
- USFWS. 2005. Biological Opinion on the proposed revision of the Endangered Species Management Plan at Fort Hood Military Installation. US Fish and Wildlife Service, Arlington, TX.
- USFWS. 2007. Black-capped Vireo (*Vireo atricapilla*) 5-year review: summary and evaluation. US Fish and Wildlife Service, Arlington, TX.
- van Nes, E. H., E. H. R. R. Lammens, and M. Scheffer. 2002. PISCATOR, an individual-based model to analyze the dynamics of lake fish communities. *Ecological Modelling* **152**:261-278.
- Walde, A., W. I. Boarman, and A. P. Woodman. 2006. Desert tortoise clearance surveys and monitoring of desert tortoises prior to translocation. RFQ171801, ITS, Barstow, CA.

- Walters, J. R. 1990. Red-cockaded Woodpeckers: a "primitive" cooperative breeder. Pages 67-102 in P. B. Stacey and W. D. Koenig, editors. Cooperative breeding in birds: long term studies of ecology and behavior. Cambridge University Press, Cambridge, UK.
- Walters, J. R., S. J. Daniels, J. H. Carter III, and P. D. Doerr. 2002. Defining quality of red-cockaded woodpecker foraging habitat based on habitat use and fitness. *Journal of Wildlife Management* **66**:1064-1082.
- Walters, J. R., P. D. Doerr, and J. H. Carter III. 1988. The cooperative breeding system of the red-cockaded woodpecker. *Ethology* **78**:275-305.
- Walther, G.-R., E. Post, P. Convey, A. Menzel, C. Parmesan, T. J. C. Beebee, J.-M. Fromentin, O. Hoegh-Guldberg, and F. Bairlein. 2002. Ecological responses to recent climate change. *Nature* **416**:389-395.
- Whitfield, M. J. and M. K. Sogge. 1999. Range-wide impact of brown-headed cowbird parasitism on the southwestern willow flycatcher (*Empidonax traillii extimus*). *Studies in Avian Biology* **18**:182-190.
- Wiens, J. A. 1989. Spatial scaling in ecology. *Functional Ecology* **3**:385-397.
- Wilcove, D. S. and L. L. Master. 2005. How many endangered species are there in the United States? *Frontiers in Ecology and the Environment* **3**:414-420.
- Wilcove, D. S., D. Rothstein, J. Dubow, A. Phillips, and E. Losos. 1998. Quantifying threats to imperiled species in the United States. *Bioscience* **48**:607-615.
- Wilkins, N., R. A. Powell, A. T. Conkey, and A. G. Snelgrove. 2006. Population status and threat analysis for the Black-capped Vireo. Department of Wildlife and Fisheries Sciences, Texas A&M University, College Station, TX.
- Wimberly, M. C. and R. S. H. Kennedy. 2008. Spatially explicit modeling of mixed-severity fire regimes and landscape dynamics. *Forest Ecology and Management* **254**:511-523.
- Wisdom, M. J., L. S. Mills, and D. F. Doak. 2000. Life stage simulation analysis: Estimating vital-rate effects on population growth for conservation. *Ecology* **81**:628-641.
- Wood, S. N. and N. H. Augustin. 2002. GAMs with integrated model selection using penalized regression splines and applications to environmental modelling. *Ecological Modelling* **157**:157-177.
- Zerger, A. 2004. GridSampler.

Appendix A. Scientific and Technical Publications

List of technical publications

Articles or papers published in peer-reviewed journals (specify whether in print, accepted for publication, or submitted for publication).

Bancroft, B.A., C. B. Wilsey, and J. J. Lawler. *In Review*. A test of single and multi-scale ensemble approaches for predicting habitat suitability. *Ecography*.

7 papers in preparation

Technical reports (specify whether in print, accepted for publication, or submitted for publication).

Shafer, S.L., J. Atkins, B. A. Bancroft, P. J. Bartlein, J. J. Lawler, B. Smith, C. B. Wilsey *In Review*. Projected Future Climate and Vegetation Changes and Potential Biotic Impacts for Fort Benning (Georgia), Fort Hood (Texas), and Fort Irwin (California). USGS open-file report. Submitted for publication.

Conference/Symposium Proceedings and/or Papers scientifically recognized and referenced (other than Abstracts).

None to report

*Published Technical Abstracts (*project-team work that used HexSim, but was not directly funded by SERDP)*

Lawler, J. J., B. A. Bancroft, and N. H. Schumaker. Potential population-level effects of climate change and land-use change on an endangered bird. *International Association for Landscape Ecology*. April 2011.

*McNyset, K., J. Falke, C. Jordan, A. Brookes, N. H. Schumaker. A spatially and temporally explicit, individual-based, life-history and productivity modeling approach for aquatic species. *American Fisheries Society*. September 2011.

Wilsey, C. B., B. A. Bancroft, J. J. Lawler, and N. H. Schumaker. Cumulative effects of cowbird management, military training, and climate-driven changes in fire regimes on the Black-capped Vireo. *SERDP Partners in Environmental Technology Technical Symposium*, December 2010.

Lawler, J. J., B. A. Bancroft, and N. H. Schumaker. The Potential Effects of Pine Senescence on Red-cockaded Woodpeckers at Fort Benning, Georgia. *SERDP Partners in Environmental Technology Technical Symposium*, December 2010.

*Huber, P. R., N. H. Schumaker, S. E. Greco, J. Hobbs. Spatially explicit population modeling and the reintroduction of a native ungulate: Using HexSim to evaluate release alternatives. *Society for Conservation Biology*, July 2010.

*Heinrichs, J. A., D. J. Bender, D. L. Gummer, N. H. Schumaker. Assessing critical habitat: Evaluating the relative contribution of habitats to population persistence. *Society for Conservation Biology* July 2010.

*Stronen, A. V., N. H. Schumaker, G. J. Forbes, P. C. Paquet, R. K. Brook. Landscape resistance to dispersal: predicting long-term effects on a small and isolated wolf population in southwestern Manitoba, Canada. *Society for Conservation Biology*, July 2010.

Wilsey, C. B., J. J. Lawler, and D. A. Cimprich. Contribution of lidar-derived measures of vegetation structure to a habitat suitability model for the black-capped vireo. *Black-capped Vireo Symposium*, 2010.

*Heinrichs, J. A., D. J. Bender, N. H. Schumaker. The relative effects of habitat loss, fragmentation, and degradation on population extinction. *International Association for Landscape Ecology*, April 2010.

*Raper, D. J., K. White, N. H. Schumaker, A. T. Morzillo. Modeling moose population dynamics under multiple climate and browsing scenarios. *International Association for Landscape Ecology*, April 2010.

Schumaker, N. H., J. J. Lawler, and A. Brookes. HexSim: A flexible simulation model for forecasting wildlife responses to multiple interacting stressors. *SERDP Partners in Environmental Technology Technical Symposium*, December 2009.

Wilsey, C. B., B. A. Bancroft, J. J. Lawler and N. H. Schumaker. The potential effects of military training activities, habitat management, and climate-induced vegetation change on black-capped vireo populations at Fort Hood, TX, *SERDP Partners in Environmental Technology Technical Symposium*, December 2009.

Schumaker, N. H., J. J. Lawler, and J. A. Heinrichs. Simulating population and stressor interactions with HexSim. *Ecological Society of America*, August 2009.

Wilsey, C. B., B. A. Bancroft, J. J. Lawler. Projected responses of black-capped vireo populations to spatial and temporal variability in human disturbances. *Ecological Society of America*, August 2009.

Wilsey, C. B. Using an Individual Based Population Model to Project the Impact of Future Climate on Animal Populations, *Region 4 Department of Defense, Environmental Protection Agency, & States Environmental*, June 2009.

*Heinrichs, J. A. and N. H. Schumaker. Population viability impacts of habitat removals: A simulation experiment with endangered kangaroo rats. *International Association for Landscape Ecology Annual Meeting*, April 2009.

Bancroft, B.A., J.J. Lawler and N. Schumaker. Forecasting the effects of land-use change on Red-cockaded Woodpeckers at Fort Benning, GA using a spatially explicit individual-based model (HexSim). *SERDP/ESTCP Annual Symposium*, December 2008.

Wilsey, C.B., B.A. Bancroft, J.J. Lawler and N.H. Schumaker. An ensemble approach to modeling Black-capped Vireo habitat for use in a spatially explicit individual-based model. *SERDP/ESTCP Annual Symposium*, December 2008.

Bancroft, B.A. C. B. Wilsey and J. J. Lawler. A multi-scale ensemble model of habitat suitability. *Ecological Society of America*, August 2008.

Lawler, J. J. and N. H. Schumaker. Modeling the role of behavior in wildlife responses to landscape change. *Society for Conservation Biology*, July 2008.

Bancroft, B.A., J. J. Lawler, N. Schumaker, and S. Shafer. Forecasting the relative and cumulative effects of multiple stressors on animal populations using a spatially-explicit individual-based model. *National Military Fish and Wildlife Association*, March 2008.

Published Text Books or Book Chapters

None to report.

HexSim User's Guide

Table of Contents

HexSim User's Guide	1
General Information	4
Author	4
Software Development	4
Contact	4
Technical Support.....	4
Financial Support.....	4
Licensing	4
HexSim Fundamentals	6
Overview	6
Interface Idiosyncrasies.....	7
Model Design.....	7
HexSim Workspaces	8
HexSim Spatial Data	9
HexSim Scenarios	10
HexSim Populations	11
HexSim Life Cycles	13
HexSim Input and Output Files.....	14
Worked Examples.....	15
Interface Basics.....	16
Interface Basics	16
The Workspace Tab	17

Overview	17
Workspace Scenarios.....	17
Workspace Spatial Data	18
The HexSim Menu.....	19
The Scenario Tabs	22
Overview	22
Simulation Parameters	22
HexSim Populations	23
The Event Sequence	24
Scenario Spatial Data.....	26
The Scenario Menu	26
The Spatial Data Viewer	30
Navigation	32
Set Data Sources	33
Creating Barrier Data.....	34
Histograms	37
Editing Hexagons	38
Editing Barriers	40
Barrier Conflicts	42
Barrier Maintenance Tools	42
Running Simulations.....	45
Population Parameters.....	49
Population Parameters	49

The Population Parameters Properties Tab.....	50
Overview	50
Initial Population Size	51
Initial Population Placement	51
Exclusion Series	52
Use Genetics	52
The Population Parameters Range Data Tab.....	53
Background	53
Range Spatial Data	54
Range Barriers	54
Range Area and Span	55
Maximum Group Members	56
Range Eligibility	57
Minimum Range Resources	57
Floater Preemption	57
Competitive Ability	58
Resource Targets	58
Importing Range Data	59
Non-territorial Species	60
The Population Parameters Traits Tab.....	61
Background	61
Accumulators.....	62
Accumulated Traits.....	64

Probabilistic Traits	64
Genetic Traits	65
Trait Builders	67
View Combinations	68
Altering Trait Data.....	69
The Population Parameters Affinities Tab	71
Background	71
Group Movement Affinities	73
Natal Affinities.....	73
Reproduction Affinities.....	73
Resource Affinities.....	74
HexSim Genetics	77
Background	77
Loci and Alleles	77
Reproduction	80
Mutation.....	82
Resource Allocation.....	84
HexSim Events.....	86
HexSim Events	86
Event Basics	87
The Accumulate Event.....	90
Background	90
Accumulator Minimum and Maximum.....	91

Temporal Smoothing	92
Accumulator Transfer	93
Allocated Hexagons.....	94
Births	94
Clear	95
Explored Hexagons	95
Group Size.....	95
Increment	96
Individual Locations	96
Mate Verification.....	98
Quantify Environment.....	98
Resources (Acquired).....	99
Resources (Explored).....	102
Resources (Competitive Acquired)	103
Resources (Competitive Explored).....	104
Stochastic Increment	105
Stochastic Trigger.....	105
Time Step	106
The Adjust Range Parameters Event	108
The Census Event	111
The Floater Creation Event.....	113
The Generated Hexmap Event	117
The Interaction Event.....	121

The Introduction Event.....	127
Movement Basics	130
The Movement Event Global Tab	135
The Movement Event Dispersal Tab.....	140
The Movement Event Exploration Tab.....	145
The Mutation Event.....	150
The Range Dynamics Event	153
The Reproduction Event	155
The Set Group Affinity Event	162
The Survival Event.....	165
The Transition Event.....	168
Analysis.....	172
Analysis	172
Census Files.....	173
Reports	175
Births	175
Deaths	176
Genetics	176
Vitals.....	177
Movement.....	177
Ranges	178
Barriers.....	179
Stochasticity	179

Interactions	180
Dispersal Steps	180
Dispersal Paths	181
Projection Matrix.....	181
Productivity	184
Tallies	185
Tally Targets	186
Tally Types.....	187
Births	187
Deaths	187
Births Minus Deaths	187
Barrier Deaths	187
Barrier Transmissions.....	187
Barrier Deflections	187
Dispersal Flux.....	187
Interactions	187
Occupancy.....	188
Trait Counts	188
Display Spatial Data	189
File Menu	189
View Menu	190
The Simulation Viewer	193
The Data Tab	193

Groups.....	194
Movement.....	195
Occupancy.....	196
The Display Tab	198
The File Menu.....	199
The View Menu.....	200
Utilities.....	201
Utilities	201
Workspace Construction.....	202
HexMap Generation.....	204
The File Menu.....	205
The View Menu.....	206
Bitmap Attributes	207
The Bitmap Parameters.....	207
The Sampling Parameters	208
The Category Data	210
Building a HexMap	212
Taking Snapshots.....	215
Using Recover	217
Global Assignment.....	218
Spatial Data and Barriers	218
Affinities	219
Log Parameters	219

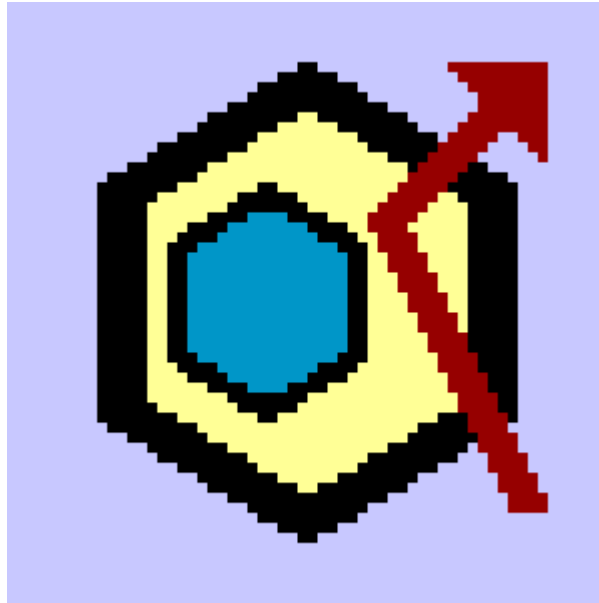
HexSim User's Guide

Descriptions.....	220
Show Usage	222
Importing Data	224
Import Scenario	224
Import HexMap Data	224
Importing Barrier Data	225
Batch Processing.....	228
Sensitivity Analysis	231
Command Line Options	234
HexSimEngine.....	234
BatchRunner.....	234
OutputTransformer	234
HexMapConverter	236
Troubleshooting	238
Troubleshooting.....	238
Common Mistakes	239
Populations.....	239
Initial Population Placement.....	239
Initial Population Resources.....	239
Exclusion Series	239
Range Data.....	239
Range Barriers.....	239
Resource Acquisition	239

Competition.....	240
Event Sequence	240
Floater Creation Events	240
Accumulate and Transition Events.....	240
Resource Acquisition	240
Event Order.....	241
Census Events.....	241
Events	241
Accumulate	241
Movement	242
Floater Creation	242
Generated HexMaps.....	242
Census.....	242
Event Outputs	243
Order Rows by Selected Traits	243
Miscellaneous.....	243
Scenario Names	243
Model Calibration.....	244
Initial Population Size and Distribution	245
Initial Trait Distribution	245
Hindcasting.....	246
Landscape Dynamics	246
Changing Disturbance Regimes	247

Simplifications and Omissions	248
Sensitivity Analysis	248
Model Performance	249
Population Trends	249
Population Distributions	250
Trait Distributions.....	251

HexSim User's Guide



HexSim User's Guide
This version was built on 04/05/2011

HexSim Users Guide

April 5, 2011

Nathan H. Schumaker
U.S. Environmental
Protection Agency
Western Ecology Division
200 SW 35th Street
Corvallis, OR 97330

HexSim is a spatially-explicit, individual-based, multi-species computer model designed for simulating terrestrial wildlife population dynamics and interactions. HexSim is generic, with landscapes, life histories, disturbance regimes, and most other details being supplied by the user at run-time. The use of HexSim is facilitated through a modern graphical user interface. HexSim is freely available.

Development of the HexSim model has been funded in part by the U.S. Environmental Protection Agency. It has been subjected to the Agency's peer and administrative review, and it has thus been approved for distribution and use. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government. The views

and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government and shall not be used for advertising or product endorsement purposes.

Preferred Citation:

Schumaker, N.H. 2011. HexSim (Version 2.1).
U.S. Environmental Protection Agency,
Environmental Research Laboratory, Corvallis,
Oregon, USA. <<http://www.epa.gov/hexsim>>

General Information

Author

Nathan H. Schumaker
US Environmental Protection Agency
200 SW 35th Street
Corvallis, OR 97330
USA

Software Development

Allen Brookes
Kevin Djang
Nathan Schumaker

Contact

Web site: www.epa.gov/hexsim
Email: schumaker.nathan@epa.gov
Phone: 541-754-4658

Technical Support

Support for HexSim will be provided on a case-by-case basis, to the extent that time is available. Contact the author if you have questions, suggestions, bug reports, or other comments.

Financial Support

The development of HexSim has been funded by both the U.S. Environmental Protection Agency and by SERDP, the Strategic Environmental Research and Development Program, through a grant (number SI-1541) to Dr. Joshua Lawler at The University of Washington.

Licensing

HexSim is not licensed, and is available free of charge.

The author, the Environmental Protection Agency, and The University of Washington assume no liability for any damages resulting from the use of HexSim.

The HexSim package includes dynamically linked libraries (dll files) associated with Syncfusion Essential Chart. The HexSim executable package was compiled using dll files from Apache (Xerces) and from Makoto Matsumoto and Takuji Nishimura (MT19937). These dll files are all licensed for redistribution by their developers. See the HexSim Licenses folder for more information.

HexSim Fundamentals

Overview

HexSim is a spatially-explicit, individual-based computer model designed for simulating terrestrial wildlife population dynamics and interactions. HexSim is very general, with landscapes, life histories, disturbance regimes, and most other details being supplied by the user at run-time. HexSim includes a sophisticated graphical user interface (GUI). The model uses spatial data to capture landscape structure, habitat quality, stressor distribution, and other types of information. HexSim can work with real or fabricated landscapes. HexSim's design makes it ideal for exploring the cumulative impacts to wildlife populations of multiple interacting stressors.

Examples of evaluations that HexSim is suited for include:

- performing population viability analysis (PVA) for one or more wildlife species;
- studying the consequences for wildlife of multiple interacting disturbances;
- assessing which habitat components are most critical for population maintenance;
- quantifying the consequences of species invasions or competition;
- designing restoration, mitigation, or reintroduction strategies;
- determining the impacts that roads and other barriers may be having on viability;
- measuring the consequences for wildlife of changes to landscape connectivity;
- exploring mechanisms linking human activities to patterns of disease spread;
- adding realism to the study of landscape genetics.

Features built into the HexSim model include the

- ability to simulate a wide range of life histories;
- ability to work with multiple interacting populations and stressors;
- ability of individual animal characteristics to change with time;
- incorporation of multiple dynamically changing landscapes;
- simulation of dynamic territory and range formation and maintenance;
- simulation of the impacts of barriers such as roads;
- use of a user-friendly interface including tools for post-processing.

HexSim simulations are built around a user-defined life cycle. This life cycle is the principal mechanism driving all other model processing and data needs. Users develop the life cycle when initially setting up a simulation. The life cycle consists of a sequence

of life-history events that the user selects from a list. This event list includes survival, reproduction, movement, resource acquisition, species interactions, and many other actions. Through the creative use of events, the user can impose yearly, seasonal, daily, or other time cycles on the simulated population. Each event can work with all, or just a segment of a population, and events can be linked to static or dynamic spatial data layers. Each life-cycle event has its own data requirements.

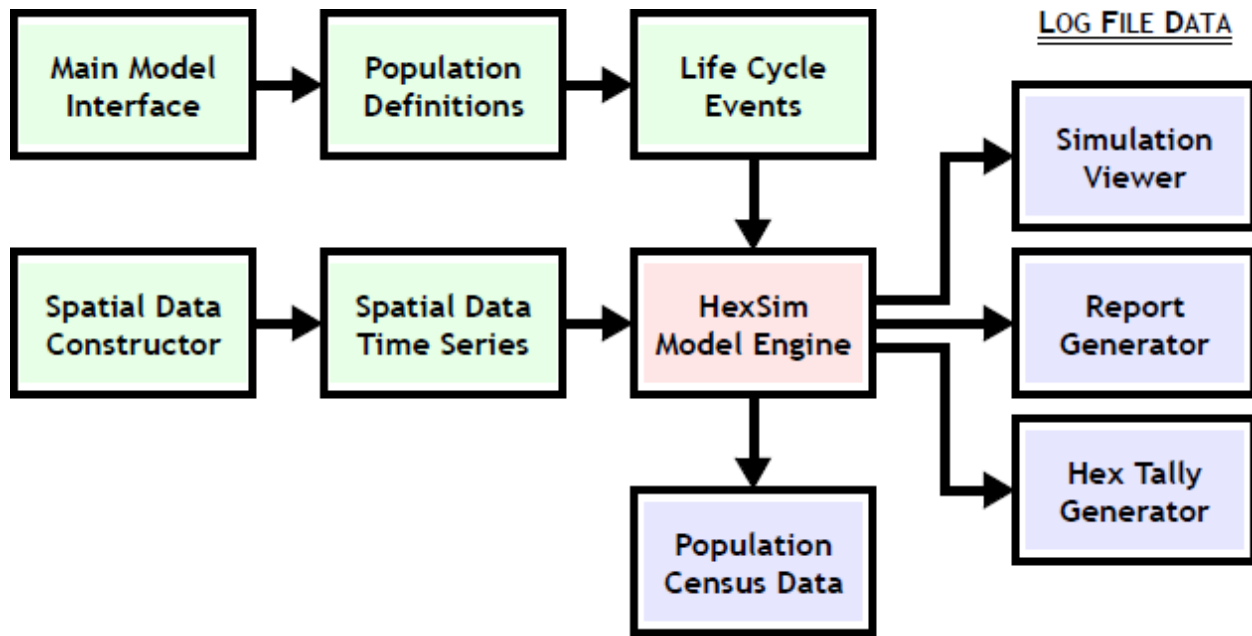
Interface Idiosyncrasies

HexSim has been used successfully on Windows XP and Windows 7, in both cases with 32-bit and 64-bit versions of the operating system. The HexSim interface was built with Windows Forms on a Windows XP platform. When HexSim is used on with Windows XP, at 96 DPI, the GUI should appear exactly as it was designed to. But when the DPI is changed, or when HexSim is used with Windows 7, some minor formatting problems may arise. The three most common problems are described below.

- Interface widgets are crowded together, or not fully resolved.
This problem is most significant on computers using a DPI setting other than 96. Windows Forms' auto-scaling feature is supposed to eliminate this problem, but it does not do so. This issue also comes up when HexSim is used on Windows 7, even at 96 DPI.
- Clicking in a check-box does not select or deselect the item.
This problem has been observed under Windows 7, in a couple of places in the HexSim interface. When a check-box does not respond to a mouse click, make sure it is selected and then hit the space bar. This should toggle the check-box state.
- The progress bar windows do not have a title.
This is a minor nuisance only observed with Windows 7. The progress bar titles can be convenient for distinguishing these windows when more than one is running at a time. These titles are not displayed when HexSim is running on Windows 7.

Model Design

HexSim is a collection of executable and other files that are integrated through a modern graphical user interface (GUI). Conceptually, HexSim consists of an interface, a model engine that conducts simulations, and a series of output processors. But HexSim also contains a wide array of tools for creating and editing spatial data, constructing and maintaining workspaces (the model's external data structures), data import and export, scenario construction, event parameterization, visualization, and so on (see figure below).

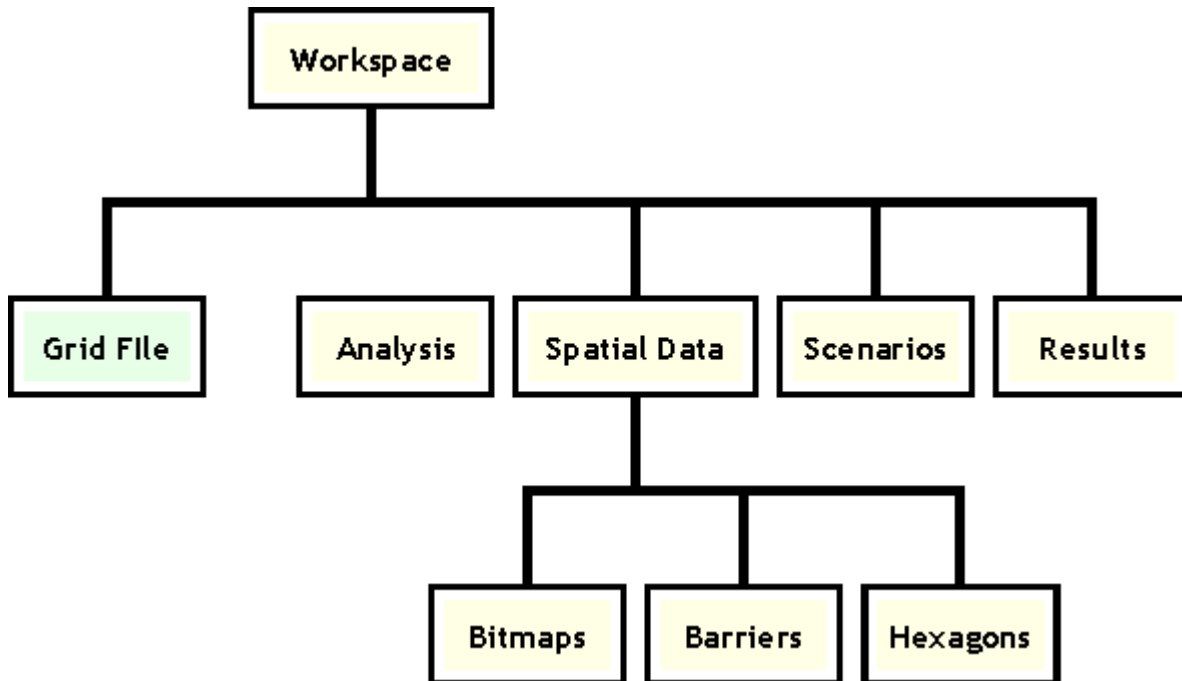


HexSim scenarios (simulation files) include descriptions of one or more populations, spatial data needs, life cycle definitions and event parametrization, and basic simulation criteria such as the number of replicates and time steps. Each population is composed of individuals, and individuals have traits that can change probabilistically, or based on age, resource availability, disturbance, competition, etc. HexSim also includes optional genetics and heritable traits. The use of traits allows individuals to have unique properties that change in time and space. Traits also allow populations to be segregated into classes, such as males and females, fitness categories, disease categories, etc. Combinations of trait values can be used to stratify events such as survival, reproduction, movement, etc.

The HexSim life cycle consists of events that are drawn from an event list. The event list has a wide array of event types, each of which can be parameterized in many different ways. Simple scenarios may use few events with minimal parameterization and little spatial data. But when more complexity is warranted, HexSim allows a great deal of data and behavior to be added to its simulations.

HexSim Workspaces

HexSim organizes projects into "workspaces". Each HexSim workspace contains a grid file, plus Analysis, Spatial Data, Scenarios, and Results folders. The Analysis folder is a place for users to add their own content. The Spatial Data folder contains additional sub-folders. Users should not alter the workspace structure. The critical elements of the HexSim workspace structure are displayed below.



The model creates and maintains the workspace structure for the user. The model interface's Workspace tab provides a view into the workspace structure. Workspaces contain input files called scenarios, spatial data, results, and other items. The scenarios and spatial data are displayed in the workspace tab. When scenarios are opened, they come up in their own tabbed windows. The HexSim menu includes options for opening multiple workspaces, creating new workspaces, build new spatial data, construct new scenarios, and many other useful operations. Scenario and spatial data context menus provide additional functionality.

HexSim Spatial Data

Each HexSim workspace contains a single grid file, and one or more spatial data time series. Grid files describe the basic landscape geometry, such as the number, size, and arrangement of hexagons. The spatial data time series are organized into themes. Each such theme is composed of one or more time steps. A time step is a single file, and a spatial data time series is a collection of such files all having a similar title. HexSim spatial data can come in the form of HexMaps and Barrier maps. HexMaps capture landcover information such as habitat quality. Barrier maps can capture the influence of linear features such as roads.

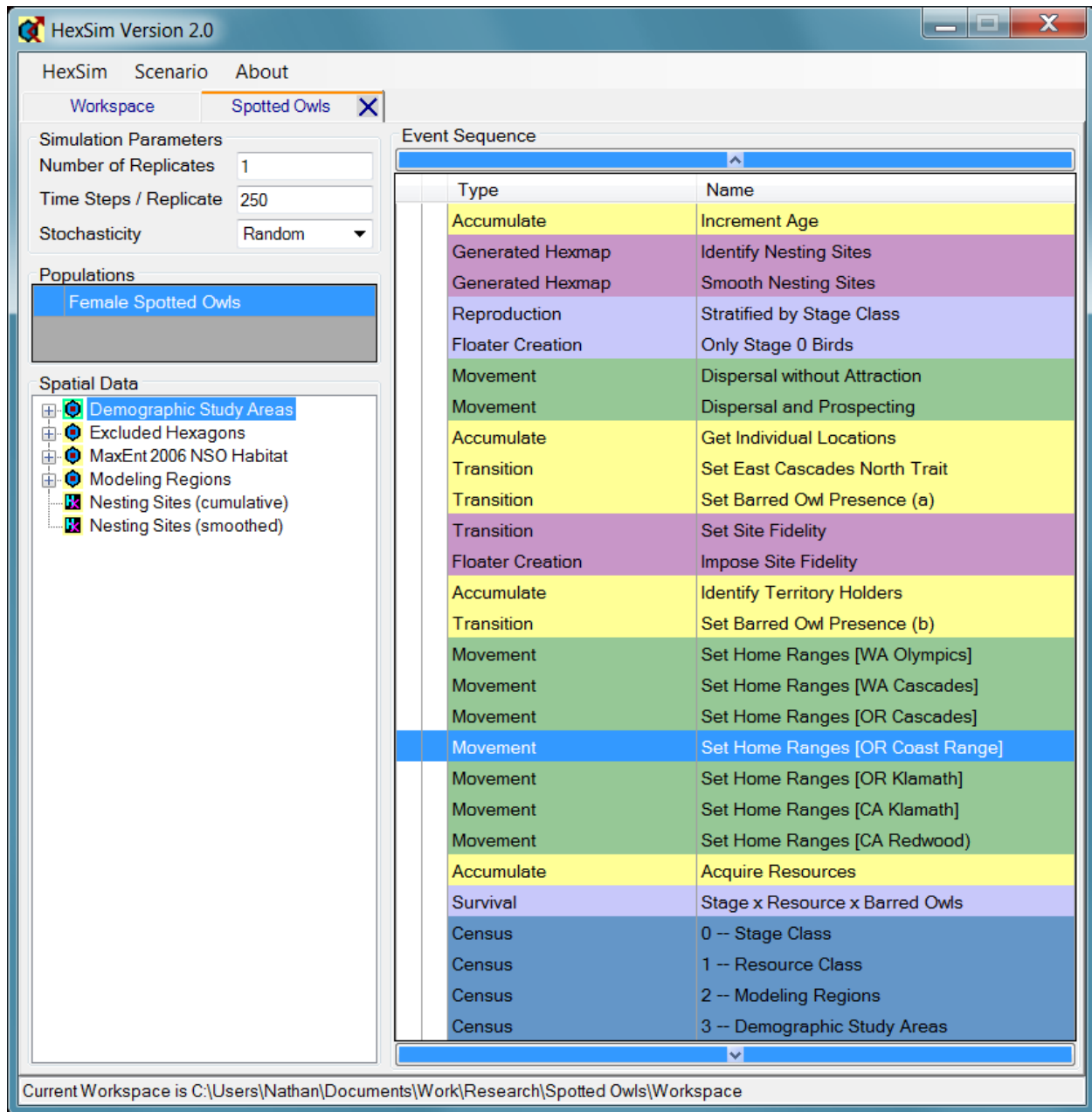
HexSim can construct spatial data time steps from comma-separated-variable (CSV) files, Shapefiles (an ESRI file format), and from 8-bit (256 color) uncompressed raster files in a Microsoft Bitmap (BMP) format. Shapefile and CSV data are imported directly, while bitmap data are intersected with the workspace grid and sampled using a tool called the HexMap Constructor. When the HexSim constructs spatial data from a bitmap, it places a copy of the bitmap file in the Spatial Data / Bitmaps folder. The

HexMaps themselves are stored in the Spatial Data / Hexagons folder, and Barrier maps are found in the Spatial Data / Barriers folder. HexMaps and Barrier maps are arranged by theme. Each theme is assigned its own sub-folder. The theme folders house one or more spatial data time steps. A HexMap time series titled "demo" consisting of steps 1, 5, and 10, would be represented by three files named demo.1.hxn, demo.5.hxn, and demo.10.hxn. All three files would be placed in a folder named "demo" within the Spatial Data / Hexagons workspace folder. HexSim will create and maintain these structures automatically.

There are two concepts to keep in mind when working with HexSim spatial data. The first is that you may use as much or as little spatial data as you like. A single HexMap is adequate for running a simulation. A more complex simulation may make use of many spatial data themes, each represented by multiple time steps. And these themes might be quite diverse, with some representing habitat, others capturing stressors, and some representing landscape barriers. The second concept to remember is that all HexSim spatial data are inherently time series. Each spatial data theme must have a HexMap representing time step one. The use of subsequent time steps is optional. When spatial data include multiple time steps, HexSim automatically inserts the correct data at the appropriate time.

HexSim Scenarios

A HexSim scenario is an XML file that contains all of the information necessary to run a HexSim simulation. Scenarios include population definitions, spatial data requirements, an event list, and event parametrization. Populations and events have sophisticated parameterization windows, but most of the model's complexity can be ignored if desired. Events can be set up to trigger once, only within a temporal window, periodically, or randomly. The model has tools for building in environmental stochasticity and for controlling density dependence. Individuals from the same, or from different populations can interact, compete for resources, share diseases, and so on. Because all of this information is stored in a single scenario file, it can be quickly retrieved and used to run a new simulation. Scenario files are small in size and easily shared. Users interact with scenarios in HexSim scenario tabs (see figure below).



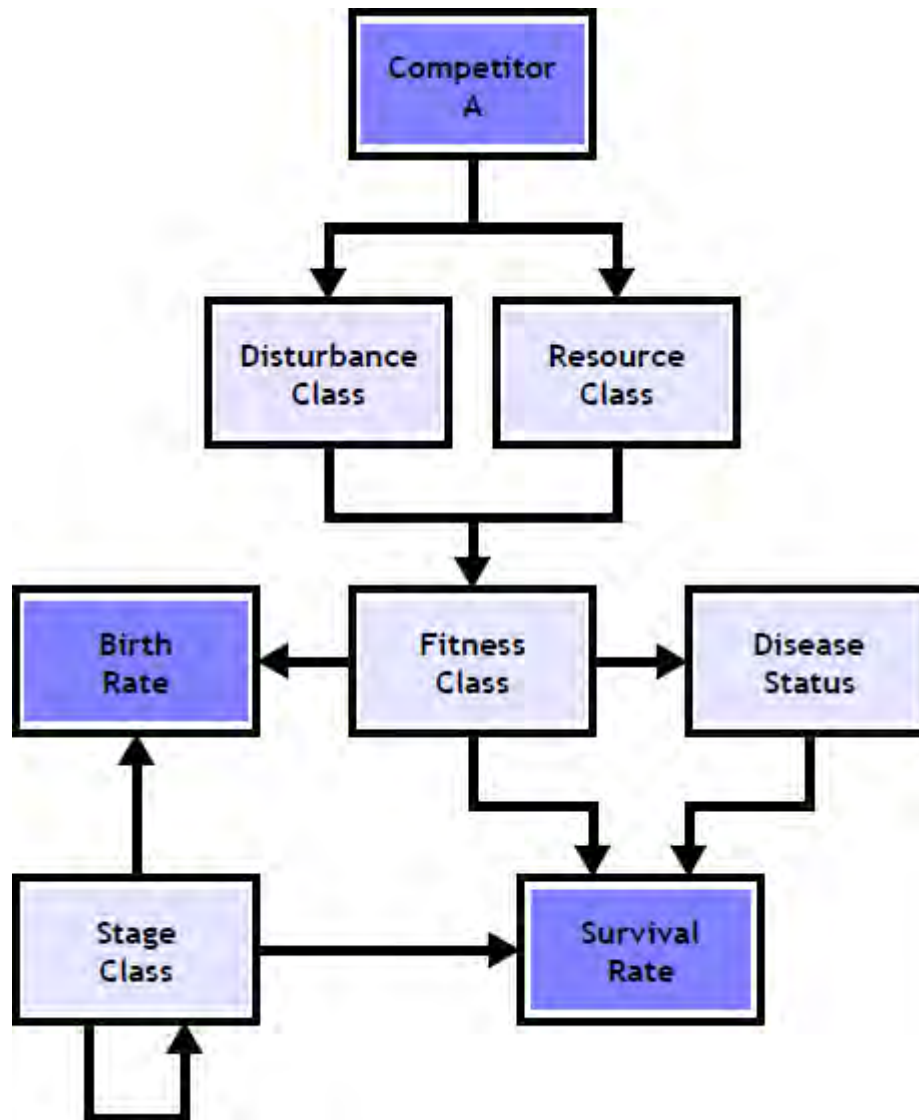
HexSim Populations

HexSim simulations must include at least one population. When multiple populations are present, individuals from the different populations may interact with each other, and they may compete for resources.

In HexSim, all individuals are placed into one of two categories -- floaters or group members. This scheme is useful for simulating territorial species, but it can also be effectively ignored when working with non-territorial animals. Group members are territory-holders, and only group members are allowed to reproduce.

The first step in creating a HexSim simulation involves setting up a population. The Population Parameter's Properties tab is used to establish some initial conditions. The Range Data tab is used to define group and range properties, use of space, resource needs and priorities, and other information. The Traits tab is where users define population traits. Every population must have at least one trait that contains at least two trait values. Trait builders have been provided to automate the creation of some of the most common trait types. The Affinities tab is used to define affinity structures. Affinities are collections of hexagons that may be revisited at some later time. HexSim has four types of affinities: natal, reproduction, resource, and group movement.

Traits are a fundamental part of HexSim scenarios. HexSim traits are defined at the population level, but they are implemented on an individual basis. What makes HexSim's traits particularly valuable is that users can set them up any way they see fit. And the traits can then be used to control most life cycle events (see figure below). Traits can influence events because events can be stratified by trait combinations. For example, a movement event might be set up to operate only on a fledgling stage class. Or a survival event might assign mortalities based on the values of a trait that reflects resource acquisition. In addition, one trait's values can also be influenced by multiple other traits, which makes it possible to set up stressor interactions and complex feedback loops. Traits can also be used to capture interactions such as parasitism, competition, mutualism, breeding, and so on.



HexSim traits can be probabilistic or accumulated (changing with time or exposure, etc.). HexSim also has a genetics module and heritable traits. Accumulated traits change based on individual experience, which is captured in a series of variables called accumulators. Events called updater functions modify accumulators, which in-turn alter accumulated traits. Transition events are used to modify probabilistic traits. Interaction events change trait values based on intra- and inter-specific interactions. Trait builders automate the process of setting up the most common trait types. Heritable traits are based on an individual's genotype, which is created at birth from its parents genotypes. HexSim genetics also include mutation, which can further impact an individual's heritable traits.

HexSim Life Cycles

HexSim simulations consist of one or more replicates, with each replicate having a fixed number of time steps. A time step is one complete pass through a user-defined life

cycle. The life cycle is made up from a sequence of events, and each event operates on one or more populations. A time step will often correspond to a year, but it does not have to. Users construct life cycles by selecting events from a list. Each event has a corresponding parameterization window that allows its behavior to be specified. The complete set of parameters associated with a simulation are stored in a file referred to as a "scenario". The model interface provides the user with a list of the scenarios available within a workspace. Scenarios can be quickly retrieved, edited, used to launch new simulations.

HexSim Input and Output Files

HexSim is a collection of files, including multiple executables. One of the executable files is the main Graphical User Interface (GUI), and another is the model engine. The role of the GUI is to construct an XML scenario file that can be passed to the engine at run-time. The scenario file holds the absolute path to the workspace being used, a list of the requisite spatial data, a description of the simulation being run, and the full set of simulation parameters.

Simulations are typically launched from the model interface, but they can also be started directly from a command window by invoking the model engine executable, and passing it a scenario file. The other executable files included with HexSim are used for a number of different tasks. Some of these can be launched directly from a command window (with specific command-line parameters), and others cannot. But every one is accessible through the main HexSim interface.

When a simulation is launched, a sub-folder is created within the workspace Results folder. This new folder is used to store the simulation output. HexSim census events create comma-separated-variable (CSV) files that track various measures of population size by replicate and time step. HexSim also stores simulation results in an log file that can be used later to generate a variety of reports, maps, and animations that illustrate the model dynamics. Interface menu options provide access to modules that automate the generation of these reports, maps, and other simulation products. HexSim also includes a dynamic simulation viewer that uses log file data to generate animated movies illustrating simulation dynamics. The use of log files as the principal repository of simulation output means that users need not anticipate what analyses they will want to perform prior to running a simulation. HexSim's log files are simply text documents with well-defined internal structure. Users have control over what data will be saved to log files. But all logging parameters are turned on by default, and users should be cautious as log files can easily grow to many gigabytes in size.

Worked Examples

A number of worked examples can be viewed on the web at www.hexsim.net.

Each of these examples is also included in the sample workspace that is bundled with HexSim. These examples illustrate a number of HexSim events and strategies. They should be a instructive for both new and experienced HexSim users.

Interface Basics

Interface Basics

[The Workspace Tab](#)

[The Scenario Tabs](#)

[The Spatial Data Viewer](#)

[Running Simulations](#)

The Workspace Tab

Overview

HexSim organizes input and output data into a collection of folders called a workspace. Workspaces contain scenarios, spatial data, and other information. The HexSim workspace tab displays the scenarios and spatial data present in the workspace that has been opened. HexSim scenarios are XML input files that fully parameterize a simulation. HexSim spatial data are time series of HexMaps or Barrier maps. Each scenario will reference some or all of the spatial data sets present within the workspace. The workspace tab is principally used to open or edit scenarios and spatial data. There are many different tools that users can access through the workspace tab, using the HexSim menu, or the scenario and spatial data context menus.

Workspace Scenarios

The workspace tab includes a HexSim drop-down menu (see below) that has options for creating scenarios, or importing existing scenarios from another workspace. Scenarios can then be opened by either double-clicking, or by using the Select this Scenario context menu item. Multiple scenarios may be opened at any one time, and each will be displayed in its own tab.

HexSim attempts to protect users by making sure that only valid scenario files can be saved. But scenarios that are works in progress can be stored in "snapshot files". The Workspace tab uses an asterisk to the left of the scenario name to indicate that a snapshot file has been created. If both snapshot and valid saved versions of a scenario exist, HexSim will always load the snapshot. Snapshot files saved in the Scenarios folder (on the hard drive) will end with a tilde (~) character.

A scenario context menu can be accessed by right-clicking on a scenario in the Workspace tab. The scenario context menu items are described below.

- **Select this Scenario**

This will open the scenario in a new workspace tab. Scenario selection can also be accomplished by double-clicking.

- **Revert to Saved Scenario**

This will remove a snapshot file and reload the saved copy of the scenario. This menu item will only be available if a snapshot file exists.

- **Delete this Scenario's Data**

This will reload the saved copy of the scenario. Any changes made to the scenario in the interface will be lost. This menu item will not be available if a snapshot file exists.

- **Display File Dates.**

This will call up a new window that allows users to remove the scenario, the scenario's results folders, or both.

Workspace Spatial Data

Workspace spatial data are all time series, and each has a name (a theme) plus one or more time steps. There are two categories of HexSim of spatial data: HexMaps and Barrier maps. HexSim uses a construct called a "Tree View Object" to list the available spatial data. Thus each spatial data time series has a little + to its left, which if clicked opens the series to display its content. The content consists of a list of one or more time steps, indicated by a green arrow and time step number. Every spatial data time series must at minimum contain a HexMap or Barrier map for time step one. The spatial data corresponding to time step one will be used for all simulation time steps unless the user supplies additional data corresponding to subsequent time steps. During a HexSim simulation, the model will automatically use the spatial data that corresponds to the current time step. For example, suppose a scenario references a spatial data time series called Habitat, and suppose Habitat consists of 5 HexMaps corresponding to time steps 1, 10, 20, 30, and 40. If the simulation is run for 50 years, then it will use Habitat time step 1 for simulation time steps 1-9, Habitat time step 10 for simulation time steps 10-19, Habitat time step 20 for simulation time steps 20-29, Habitat time step 30 for simulation time steps 30-39, and Habitat time step 40 for all remaining simulation time steps.

The HexSim interface provides tools for reorganizing and renaming spatial data..A spatial data time series is really just a sub-folder that resides in the Workspace Spatial Data folder. And the individual time steps are just files with a "hxn" (HexMap) or "hbf" (HexSim Barrier Format) extension that reside within the time series sub-folder, and that follow a specific naming convention. Individual time steps are named using the series title as a prefix and their step number as a suffix. For example, a spatial data time series of HexMaps named Habitat, with time steps 1, 10, and 20, will correspond to the a folder named Habitat in the workspace Spatial Data location. Within the Habitat folder, there would then be three HexMaps named Habitat.1.hxn, Habitat.10.hxn, and Habitat.20.hxn.

Spatial data context menus can be accessed by right-clicking on a spatial data time series or on an individual time step. The context menus change depending on whether the target is a series or a time step, and if it is a HexMap or a Barrier map. These context menus provide access to features such as: display, save as, copy, edit, rename, delete, etc. The workspace tab context menu items are described below.

- **Copy**

Applies to HexMap and Barrier map series. Copies the selected series.

- **Rename**

Applies to HexMap and Barrier map series. Renames the selected series.

- **Delete**

Applies to HexMap and Barrier map series and time steps. Deletes a series or time step.

- **Alter Hexagon Values**

Applies to HexMap series and time steps. Changes hexagon scores for a HexMap series or time step.

- **Save As**

Applies to HexMap series and time steps. Saves series or time steps as CSV files, Shapefiles, or bitmaps.

- **Display**

Applies to HexMap and Barrier map time steps. Displays the selected time step.

- **Copy Time Step**

Applies to HexMap and Barrier map time steps. Copies a HexMap or Barrier map time step.

- **Rename Time Step**

Applies to HexMap and Barrier map time steps. Renames a HexMap or Barrier map time step.

- **Scale Hexagon Values**

Applies to HexMap time steps. Scales hexagon scores for a HexMap time step

- **Smooth Hexagon Values**

Applies to HexMap time steps. Runs a moving window filter through a HexMap time step. See the [Generated HexMap](#) event for more details.

- **Edit Attributes**

Applies to Barrier map time steps. Changes Barrier map attributes.

The HexSim Menu

Workspace-specific functions are located within the HexSim drop-down menu. The HexSim menu is used to accomplish a range of different tasks including loading

workspaces, working with batch files, importing data, constructing HexMaps, and so on. These HexSim menu contains the items listed below.

- **Set Workspace**
Select and load a workspace in the current instance of HexSim.
- **Recent Workspaces**
Select a recently visited workspace in a new instance of HexSim.
- **Add Workspace**
Open a new instance of HexSim.
- **Add Scenario**
Create a new blank scenario.
- **Copy Workspace**
Duplicate an existing workspace.
- **Create Workspace**
Create an entirely new workspace.
- **Edit the Batch File**
Opens the Batch File Editor.
- **Import Scenario**
Copy a scenario from one workspace into another.
- **Import Barrier Data**
Import a HexSim Barrier map from a workspace or data file.
- **Import HexMap Data**
Import a HexMap from a data file.
- **Simulation Viewer**
Start the HexSim Dynamic Simulation Viewer.
- **Generate HexMap**
Start the Generate HexMap tool.
- **Display Spatial Data**
Display output data files as HexMaps.

- **Set HexMap Colors**

Allows users to choose between several different color ramps. The color ramps are used to display relative hexagon quality.

- **Grid Info**

Show the grid attributes.

- **Stop Dependent Processes**

Kills all child processes launched from the main GUI.

- **Exit**

End this HexSim session.

The Scenario Tabs

Overview

The HexSim workspace tab contains a list of the scenarios present in the workspace. When a scenario is selected (by double-clicking, or by using the context menu), it opens up in a scenario tab. Multiple scenarios can be opened simultaneously, and each will appear in its own tab. Each scenario tab contains a set of simulation parameters, a list of HexSim populations, an event sequence, and a list of the scenario's spatial data. Also, the Scenario drop-down menu provides users with access to a large number of important functions. These constructs are discussed below. Different scenarios need not have any similarities except that they must all reference the parent workspace's spatial data.

Simulation Parameters

There are only three simulation parameters: The number of replicates, the number of time steps per replicate, and the stochasticity model. Each simulation will consist of one or more replicates, and each replicate will have the same number of time steps. The stochasticity model is only relevant in special circumstances. Specifically, survival, reproduction, transition, and mutation probabilities may be supplied in the form of a collection, with one member of each collection being selected per time step. These three event types need not use this feature. But if a collection of probabilities is supplied, the expectation is that this is being done in order to simulate environmental stochasticity. The stochasticity model specifies how members are selected from the collections.

When simulating environmental stochasticity, the most complicated issue to keep track of is the correlation between events. For example, a good year for survival is likely to also be a good year for reproduction, and visa-versa. HexSim has addressed this issue through the adoption of a couple of policies. Survival, Reproduction, Transition, and Mutation events may each have a single set of probability values, or a collection of values. But all collections must have a fixed size. The collection size is not a constant (e.g. it can be 100 or 1000), but any given simulation must use a single fixed collection size. Then, HexSim will set an index value (the stochasticity index) at the start of each time step and use this index to draw an element from each collection that has been supplied. This only has relevance for Survival, Reproduction, Transition, and Mutation events that are being used to simulate environmental stochasticity, since otherwise the collection size is one. Users may include events with a collection size of one, and with a collection size exceeding one in the same simulation.

The stochasticity model is used when HexSim sets the stochasticity index at the beginning of each time step. Suppose a scenario has a Survival and a Reproduction

event that are both being used to simulate environmental stochasticity. Suppose also that the collection size is 100, with the items in the collection being numbered 1, 2, 3, ..., 100. If the stochasticity model is set to Random, then the stochasticity index will be selected by drawing a random variate from $U[1, 100]$, where U represents a uniform distribution. If the stochasticity model is Cyclic, then the index will step incrementally from 1 to 100, and then will wrap back to 1 again. This single stochasticity index is used to for every stochastic event present in the event sequence.

Thus HexSim's approach to stochasticity is very general. It allows users to set up any amount of correlation between events. Stochasticity will be correlated between events when correlation exists between members of the various collections of vital rates. And since the collection size may be arbitrarily large, this scheme can be effectively used to simulate random fluctuations. Conversely, users may supply very small collection sizes -- for example a collection of four elements representing three good years and one bad year. But with this flexibility comes a cost: the burden of creating the collections of stochastic vital rates falls entirely on the user. It will take some effort to assemble a collection of survival rates and a collection of reproductive rates that are ordered so as to capture correlations between good and bad time steps.

HexSim Populations

All HexSim simulations involve one or more populations. Each population has a set of unique parameters, but all populations utilize a common event sequence. Population-specific parameters are collected in the Population Parameters window. Operations including create, edit, rename, and delete are accessed through the population panel's context menu. The Population Parameters window allows users to define initial conditions, an exclusion layer (e.g. a map of oceans and lakes), resource needs, use of space, competitive ranks, individual traits, and movement affinities. Heritable traits can also be developed if the genetics sub-model is turned on.

When a new population is added, HexSim will also add a Movement event for the following reason. When the scenario starts up, the initial members of the population are placed into the landscape as floaters. The Movement event that is automatically added exists simply to assign these floaters some resources. By default, this Movement event employs an Exploration Only strategy, and the exploration goal is set to Start new else join Existing. The intent is that the floaters will simply look around them and see if they can start or else join a group. If so, they will have access to range resources. If not, they will end up with a Floater Allocation. Naturally, users may customize this Movement event, or even delete it.

The Movement events that are automatically added to the life cycle when a population is created are referred to as "locked". They are locked because they serve an important function and should not be removed carelessly. These automatically generated Movement events are the only events in HexSim that are locked. The significance of the event's locked status is that it cannot be moved from the top of the event sequence. In addition, locked events are set to trigger only at time step 1, and this triggering cannot

be changed. Finally, a warning message is displayed if users try to remove locked events (removing locked events is allowed). Locked events are colored black as a reminder to users. Experienced users will frequently alter or remove these locked events.

The Event Sequence

The Event Sequence is a list of actions that the model will perform during each time step. The individual actions are called "events". The event sequence is assembled by selecting events from a context menu drop-down list. This is found by right-clicking in the event sequence, and then using the Create a New Event sub-menu. The order of the events can be changed using up and down arrows located above and below the event sequence. Each event has a corresponding parameterization window, which can be opened by double-clicking the event, or through its context menu. The event sequence and parameterization data are stored in HexSim scenario files.

A HexSim simulation is comprised of one or more replicates, and each replicate is a collection of time steps. The actual length of a time step (e.g. day, season, year) is defined implicitly based on the event sequence. The number, order, and parameterization of events assembled by the user collectively make up the core of a HexSim simulation. There is a single event sequence per scenario. If a scenario includes multiple populations, then the events for each population should be interspersed throughout the event sequence. The event sequence context menu provides access to a useful collection of additional tools. These tools are listed below.

- **Create a New Event**
Add an event to the event sequence.
- **Edit the Selected Event**
Open the event parameterization window.
- **Copy the Selected Event**
Copy an event.
- **Rename the Selected Event**
Rename an event.
- **Delete the Selected Event**
Delete an event.
- **Show Usage for the Selected Event**
Invoke the Usage Summary tool and load it with this event's data.
- **Set the Selected Event's Display Color**

Set the color used to display the event in the event sequence.

- **Set All Display Colors**

Set all events to the same color.

- **Clear the Selected Event's Display Color**

Clear this event's color specification.

- **Clear All Display Colors**

Clear all events' display colors.

- **Set the Selected Event's Triggers**

Specify the time steps for which an event will be active (triggered).

- **Clear the Selected Event's Triggers**

Clear this event's triggering data (thus making it trigger for all time steps).

- **Clear All Event Triggers**

Clear all events' triggering data.

- **Display Population Names**

Show a column of population names in the event sequence.

- **Display Event Types**

Show a column of event types in the event sequence.

- **Display Event Names**

Show a column of event names in the event sequence.

There are a few cases in which an event is added by HexSim itself. For example, when a new population is created, a Movement event is added to the event sequence. This event set up to run only at model initialization, and it exists strictly to assign resources to the starting population. These automatically added Movement events are colored black, and they are referred to as locked. An extra warning message must be dismissed before a locked event can be deleted. HexSim Trait Builders also add events for the user. These events are distinguished with a gray color, and they are not locked.

HexSim events normally trigger once every time step,. But events can be made to trigger only at select time steps. Event triggering is controlled through context menu options. Right-clicking on a HexSim event will bring up its context menu. A slightly more advanced feature (involving accumulated traits and updater functions) can be used to make events effectively trigger randomly.

In addition to creating events, and setting up triggering, the event sequence context menu can be used to edit, copy, rename, and delete events. It can also be used to set event colors, and to control the content displayed in the event sequence.

Scenario Spatial Data

The Scenario Spatial Data window provides a list of the spatial data used by the events in the event sequence. A scenario's events may use some or all of the spatial data available in the workspace. Most actions performed on scenario spatial data are initiated using context menus. The context menu items associated with the workspace and scenario spatial data differ. The scenario spatial data context menus allows users to display, set up cycling, and to toggle an ignored state flag. The cycling option is accessed by right-clicking on the time series. The display and toggle ignored state options are accessed by right-clicking on an individual time step.

Cycling is used to define a time step at which the model will loop back to the beginning of a spatial data time series. If the cycling context menu item is selected, then a window will pop-up that allows the cycle length to be set. A cycle length of zero is used to indicate that cycling has been turned off. Non-zero cycle lengths set the number of time steps for which HexSim will proceed normally through the spatial data. After completing the time step whose number equals the cycle length, HexSim will cycle back to time step 1.

A HexSim scenario can also be instructed to ignore specific spatial data time steps. The context menu item called toggle ignored state makes this possible. When a time step has the ignore flag set, a red X appears over the green arrow icon. Toggling the state of an ignored time step removes the red X and makes the time step active again.

The Scenario Menu

Scenario-specific functions are located within the Scenario drop-down menu, at the top of the HexSim GUI. This menu provides access to many critical functions such as saving and renaming scenarios, creating batch files, generating tallies and reports, running simulations, and more. These menu items are listed below.

- **Save**

Saves the current scenario to the disk. Only valid scenarios may be saved. Invalid scenarios (works in progress) may be saved by taking a snapshot (see below). This item has a keyboard shortcut: Ctrl+S.

- **Save As**

Saves the current scenario with a new name.

- **Save to Batch File**

Adds the current scenario to the list of scenarios that will be processed when HexSim is run in batch mode. There can only be a single batch file per workspace, and its location is always at the top of the workspace.

BatchRunner.exe is the tool that runs HexSim in batch mode. This executable can be passed zero, one, or two arguments. If the first argument is an integer, it will be used to indicate the number of simultaneous processes that should be launched (and maintained). For example, on an 8-core system, one may want to run 7 simultaneous simulations, leaving one CPU for the operating system. If this parameter is missing, the number of simultaneous processes defaults to 1.

BatchRunner's other command-line parameter is the path to the workspace whose batch file is to be executed. If this parameter is missing, then BatchRunner will invoke a file-chooser, which will allow users to navigate easily to the desired directory.

- **Rename**

Changes the name of the current scenario.

- **Take Snapshot**

Makes a disk copy of a scenario without overwriting the saved copy. Snapshots allow the user to store changes without losing the original copy. Snapshots are also useful when scenario editing is only partly complete, since invalid scenarios cannot be written to the disk using the Save or Save As tools. Scenarios and scenario snapshots have identical disk file names except that snapshots end with a tilde character.

This item has a keyboard shortcut: Ctrl+T.

- **Recover**

Reloads the saved copy of an event or parameterization window. The Recover sub-menu is broken into groups. At the top is an option for recovering the simulation parameters. Next, an additional group of recover options is dedicated to each population. Any Generated HexMap events will be placed at the bottom. At the top of the population-specific sub-menu's is an item with the population's name. This option allow you to recover the population parameters. Finally, each event assigned to the population is listed, and can be selected to recover just the parameters associated with that event.

- **Recover All**

Recovers every scenario parameter from the saved copy on the disk. This is equivalent to reloading the scenario from within the Workspace tab.

- **Global Assignments**

Global assignment is available for spatial data (including Barriers), affinities, log parameters, and event descriptions. Global assignment allows users to quickly review or modify all instances of a data class. For example, many events may make use of spatial data. Changing the assignment of spatial data throughout a complex scenario would otherwise necessitate opening and altering many event parameterization windows. The global assignment tool allows these changes to be made from within a single window.

- **Show Usage**

This menu item calls up the Usage Summary window. The usage summary allows users to quickly see all of the dependencies built into a simulation. These include spatial data, barrier data, traits, accumulators, affinities, and genetics. The usage summary window groups items by color. When a cell from the Item or Parameter column is selected, the grouping is shown across all columns in the table. When a cell from the Current Value column is selected, the grouping is made just of like items within that column.

- **Generate Tally Data**

This tool uses simulation results stored in log files to generate hexagon-by-hexagon tallies for various measures of population performance. The tallies can be saved as HexMaps or CSV files.

Tally types include Births, Deaths, Births minus Deaths, Barrier Deaths, Barrier Transmissions, Barrier Deflections, Dispersal Flux, Interactions, Occupancy, and Trait Counts. Users specify the range of time steps for which the tally data will be collected. HexSim then walks through the log file and for all time steps within the specified range, and it tallies up occurrences of the specified tally type.

The tally target can be set to Groups, Floaters, or Groups and Floaters. If the tally target is groups, then the group data is averaged across each of the range hexagons. Floaters do not hold ranges, so if the tally target includes floaters, then each data point is attributed to a single hexagon from the range (group members) or explored area (floaters).

The tally data file is placed in the scenario's results folder. These data can then be displayed using the Display HexMap tool, and CSV files can be loaded into a spreadsheet.

- **Generate Report**

This tool can generate a CSV report from log file data. Report types include: Births, Deaths, Vitals (observed vital rates), Movement, Ranges, Barriers (barrier interactions), Stochasticity (a time series of stochasticity indices), Interactions (data from interaction events), Dispersal Steps (frequency distribution), Dispersal Paths (frequency distribution), Dispersal Successes (frequency distribution), and a

Projection Matrix (observed, stratified by traits. The report file is placed in the scenario's results folder.

- **Generate Census Files**

HexSim Census events automatically generate CSV files. This menu item uses log file data to construct a parallel set of CSV output files.

Census events allow users to select which trait combinations are to be included in the output file. This is convenient because only a subset of the possible trait combinations may be of interest. When census data is generated using this HexSim menu item, the resultant CSV file contains all possible trait combinations.

The Census event output data includes a column for each selected trait combination, and the event itself provides a key to these files. The View Combinations tool, which is accessed from the Population Parameters Traits and Accumulators tab, can be used to interpret the census data files generated from log files.

- **Run Simulation**

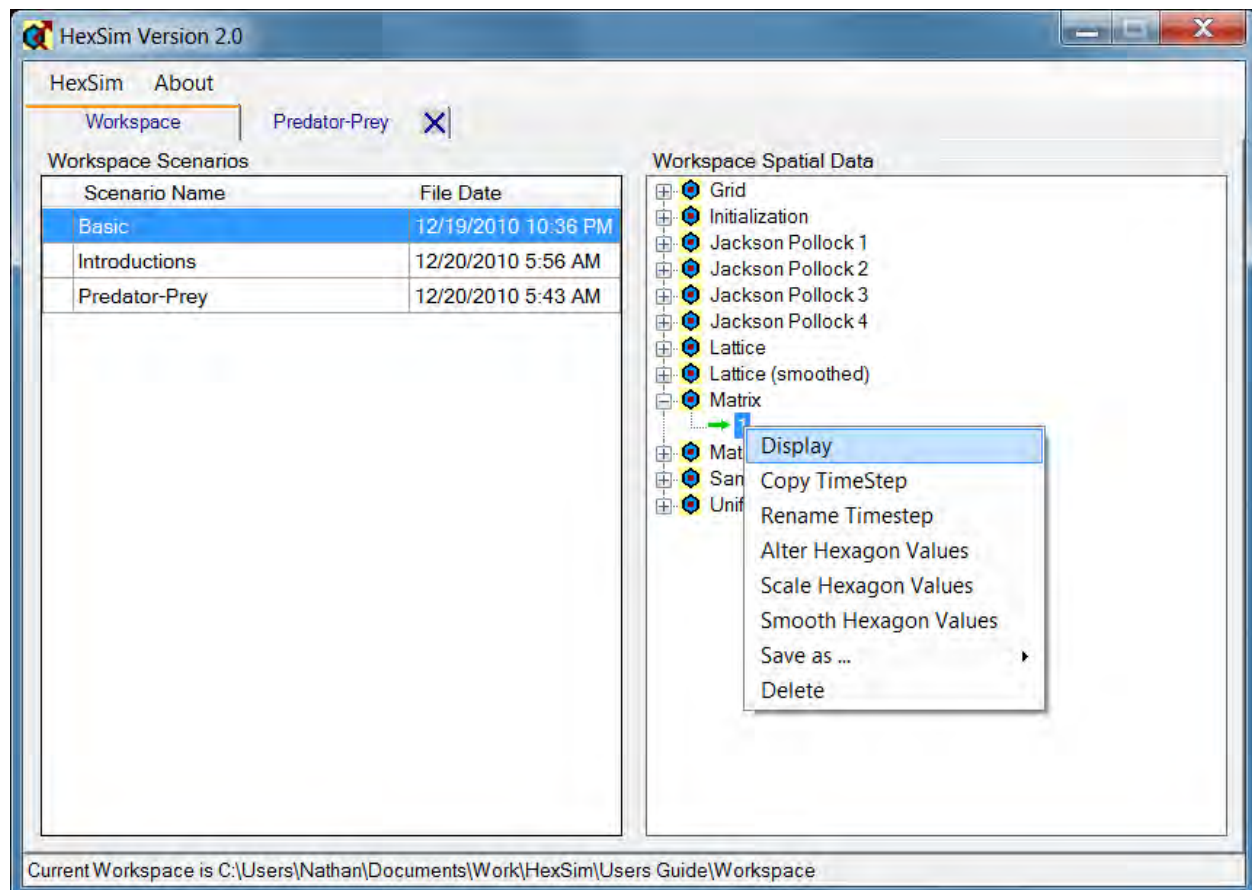
Calls up the HexSim Monitor window, which allows you to run a simulation. The Monitor tool has Text, Errors and Chart tabs. It also allows users to select checkboxes for close upon completion, using a fixed random number generator seed, and for saving results in a time-stamped directory. Time stamped directories are particularly useful when going back and forth between running and viewing simulations, using the dynamic simulation viewer. Windows file-locking prevents users from overwriting simulation data when those results are open in the dynamic viewer.

- **Close Tab**

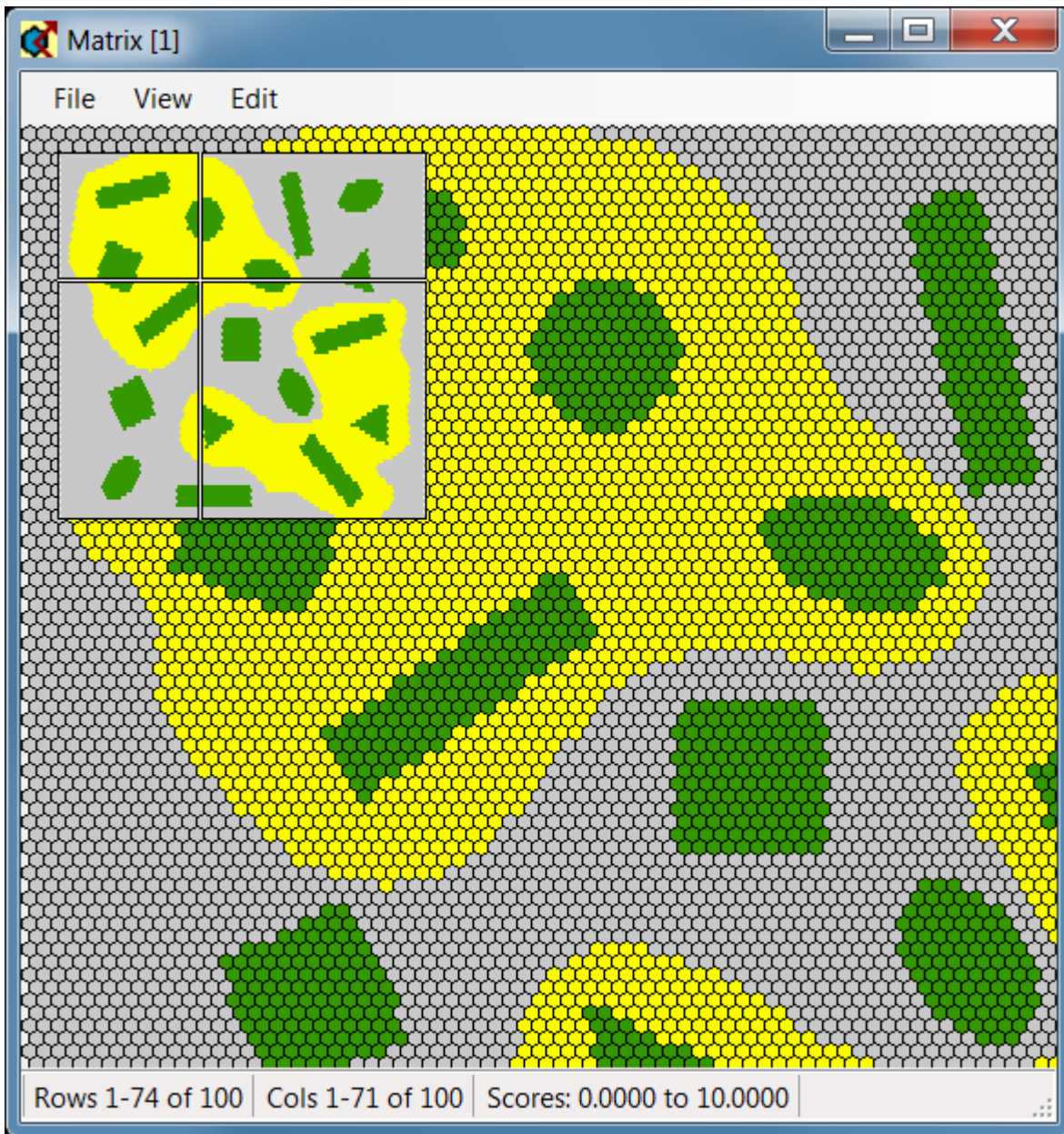
Closes the current scenario tab. Any changes that have been made since the last Save, Save As, or Snapshot will be lost.

The Spatial Data Viewer

HexSim includes a tool for viewing HexMaps and Barrier maps. The Spatial Data Viewer can be accessed from the HexMap and Barrier map context menus (see image below), or by simply double-clicking on a spatial data time step.



In addition to providing a display of a HexMap or Barrier map, the Spatial Data Viewer provides File, View, and Edit menus, a Status Bar, and a Navigation Window. These features are all shown in the figure below.



The File, View, and Edit menus offer the following options

File

- Set Data Sources
- Create New Barrier Data
- Show Histogram

View

- Zoom In and Out

- Show the Navigation Window
- Show the Grid Lines
- Show Barrier Edges

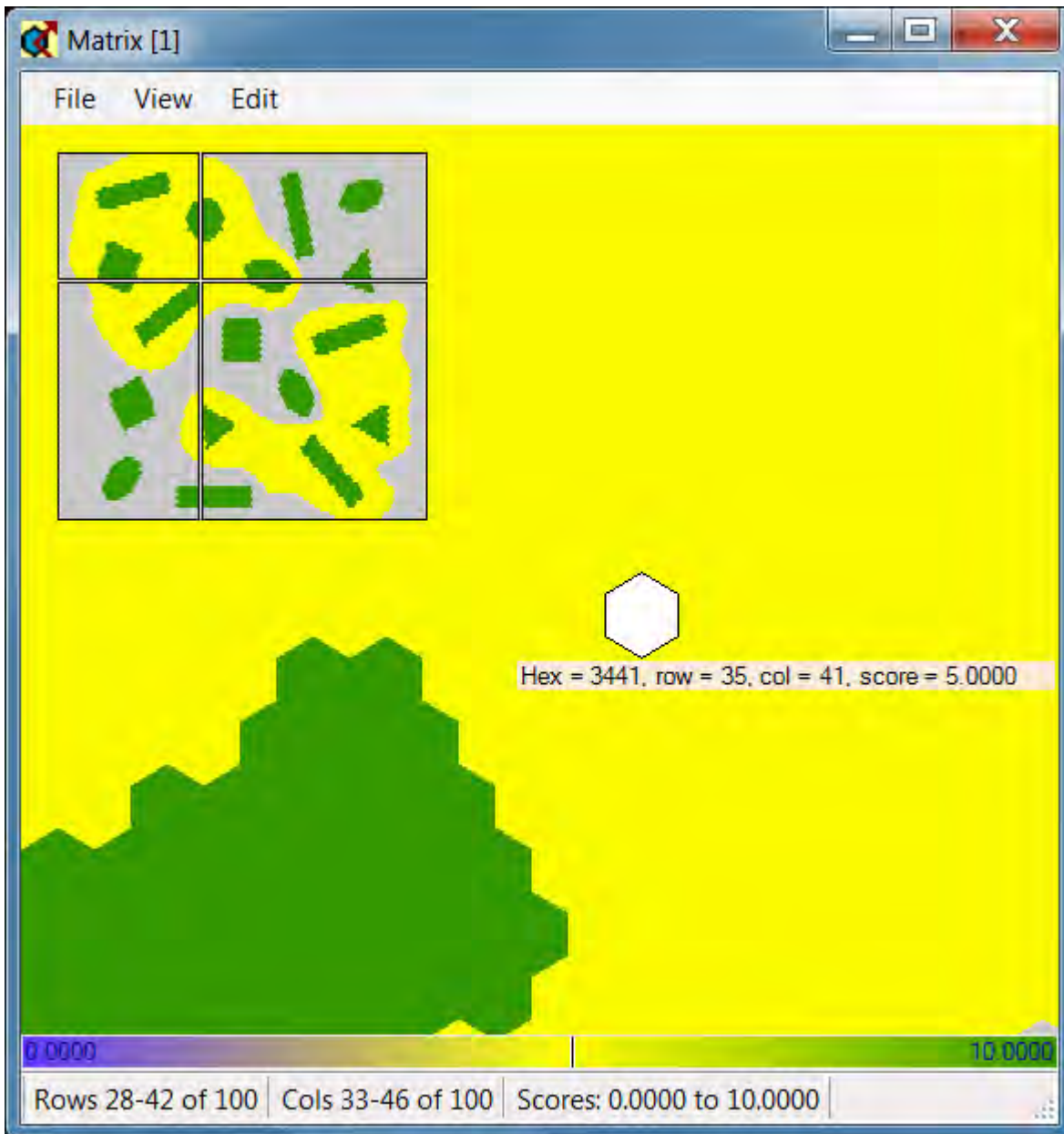
Edit

- Edit Hexagons
- Edit Barriers
- Edit Barrier Attributes
- Add Barrier Category Pair
- Delete Barrier Category Pair
- Set Current Barrier Category Pair
- Compact Categories

These functions are described in the sections below.

Navigation

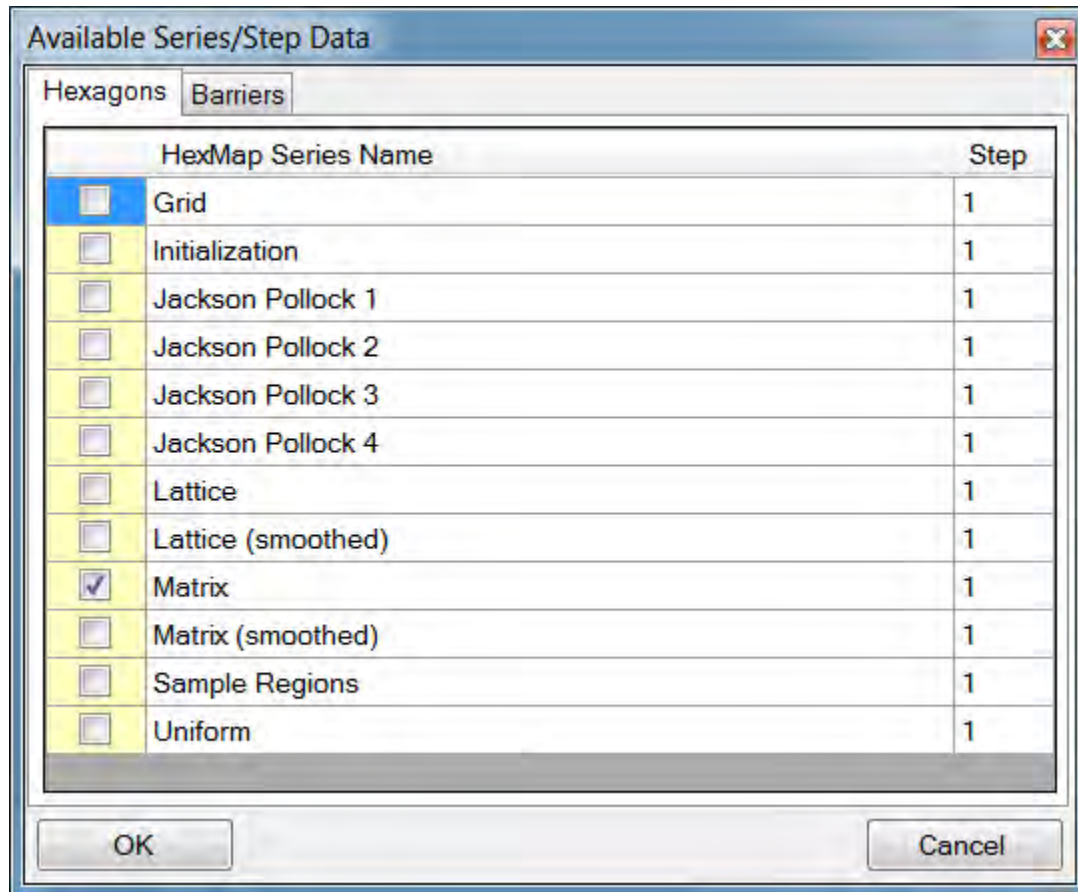
The Spatial Data Viewer has a Navigation Window that can be used to quickly move about in the display (see figure above). The cross-hairs indicate the midpoint of the region currently being displayed. Hexagons can be selected by clicking on them, and doing so displays their ID, row, column, and score in a tool tip. In addition, the color ramp being used is displayed along the bottom of the window, with a vertical bar that indicates the score of the selected hexagon, relative to the range of scores present in the entire spatial data time series. The range of scores presented in the status bar are gathered from the HexMap being displayed. The range of scores indicated in the color ramp (visible when a hexagon is selected) are gathered from the collection of HexMaps making up the time series. Together these two sets of information indicate how the selected hexagon ranks within the current HexMap, and how the current HexMap ranks among the spatial data time series of which it forms a single time step. This is illustrated in the figure below.



The View menu has options for zooming in and out of the spatial data. In most cases, the + and - keyboard shortcuts for zooming will be much more useful. There is also an option for toggling the Navigation window on and off. But this can be accomplished by simply right-clicking on any hexagon. The View menu has options to turn the grid lines on and off, and to toggle the display of barriers edges. The latter option will be inactive unless barriers have been loaded into the tool. Barriers can be loaded into the Spatial Data Display by simply displaying a Barrier map, or by using the Set Data Sources tool, which is present in the File menu. When a hexagon is double-clicked, HexSim will attempt to center it in the Spatial Data Viewer.

Set Data Sources

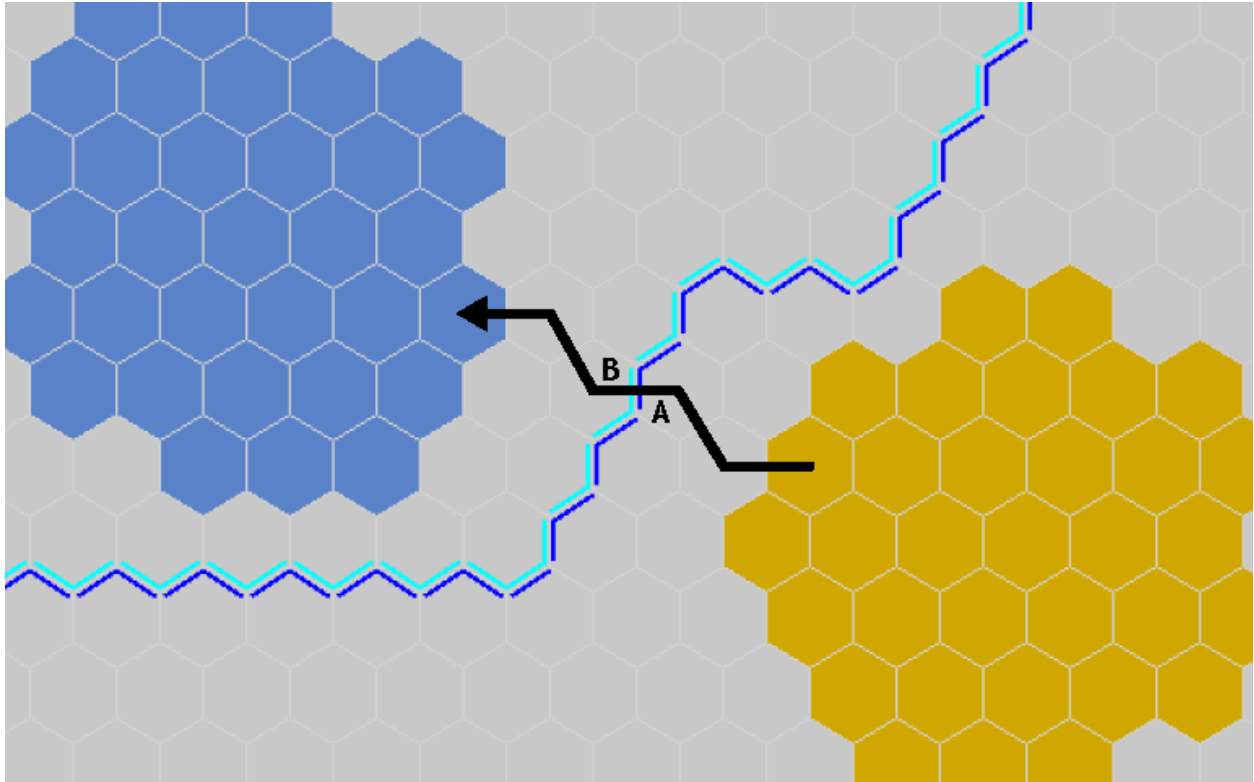
The File menu item named Set Data Sources allows users to select a HexMap and a Barrier map to display in the Spatial Data Viewer. One map of each time can be displayed at once. When this option is selected, the Available Series / Step Data dialog window comes up. Separate tabs are devoted to HexMap and Barrier map selection. An image of this dialog window is shown below.



When a Barrier map is displayed, the grid lines will be changed from black to gray. This makes it easier to see the barrier segments.

Creating Barrier Data

HexSim barriers are bidirectional, and thus come in pairs. Each side of a barrier pair can have its own distinct properties. Barriers have probabilities for mortality, deflection, and transmission. Together these probabilities must sum to 1.0. Barriers associated with a hexagon represent the **cost of exit** from that hexagon. For example, in the figure below, a blue barrier edge is encountered upon exit from hexagon A. In the same step, a cyan barrier edge is crossed upon entry into hexagon B. In this situation, the blue barrier edge will impact the disperser. There will be no consequence of crossing the cyan edge. Had the individual moved the other direction, then the cyan edge would have influenced it instead. This distinction can be ignored if both edges are assigned identical properties.



There is an option called Create New Barrier Data in the File menu. This option allows users to develop a new Barrier Map by hand. When it is selected, users must first supply a name and time step for the new Barrier Map. Next, an Edit Barrier Attributes dialog window comes up. This window (displayed below) is used to set the probabilities of mortality, deflection, and transmission for each member of the barrier pair. Typically, the two sides (Category 1 and Category 2 in the figure) will be set the same. The Edit Barrier Attributes window can be called up from the Edit menu as well.

Edit Barrier Attributes

Category Label: 100% Deflection

Category 1

Probability of Mortality: 0.0000

Probability of Deflection: 1.0000

Probability of Transmission: 0.0000

Category 2

Probability of Mortality: 0.0000

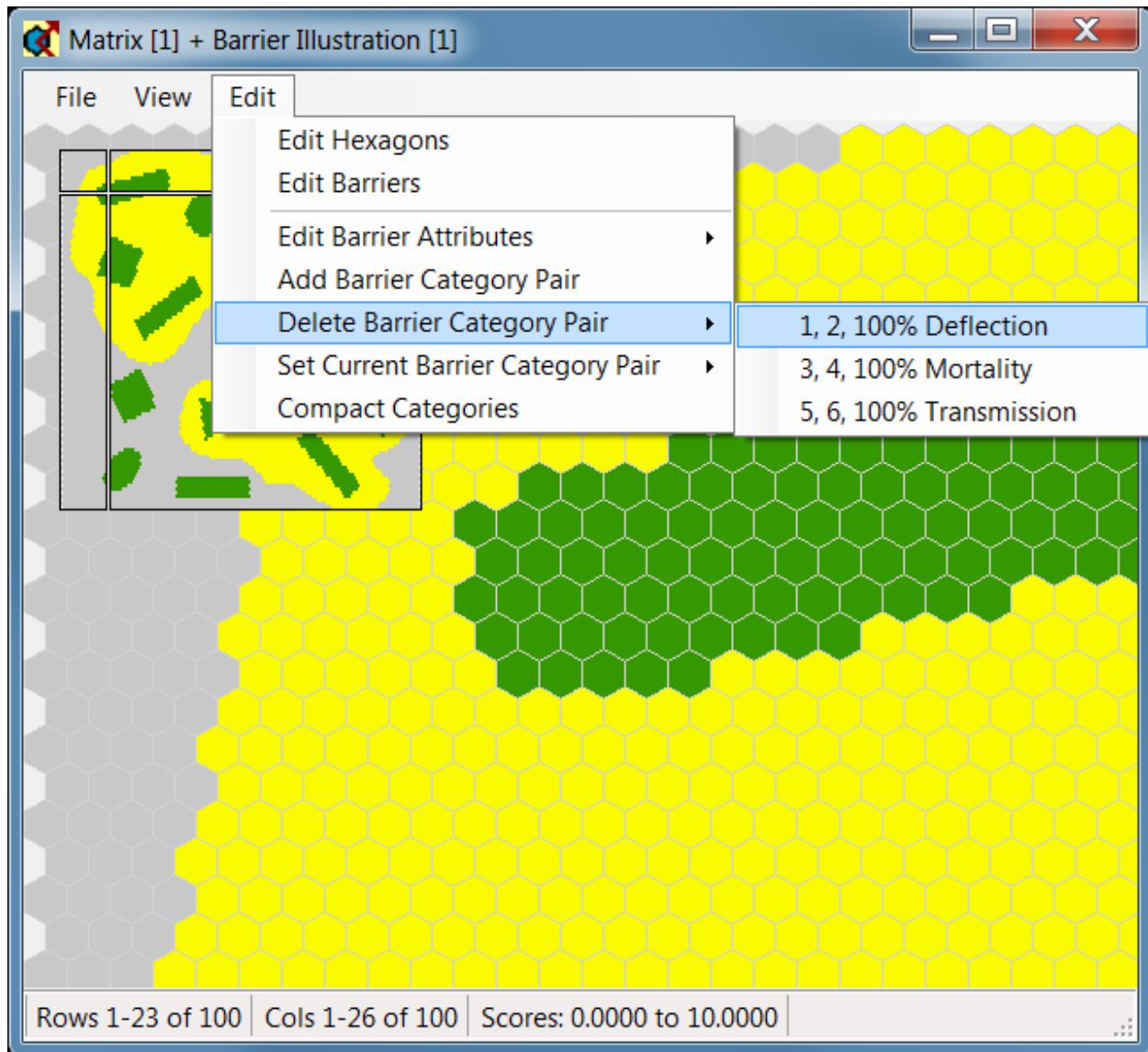
Probability of Deflection: 1.0000

Probability of Transmission: 0.0000

Clicking Accept will PERMANENTLY ALTER these Barrier Attributes BOTH IN MEMORY AND ON DISK.

Accept Cancel

The Categories in the figure above are color-coded to match the barriers displayed in the Spatial Data Viewer. Multiple barrier series can be used in a simulation, and each series may have multiple category pairs. The Category Label field provides a convenient way to associate a meaningful name with each category pair. But their use is entirely optional. The figure below illustrates the use of category labels.



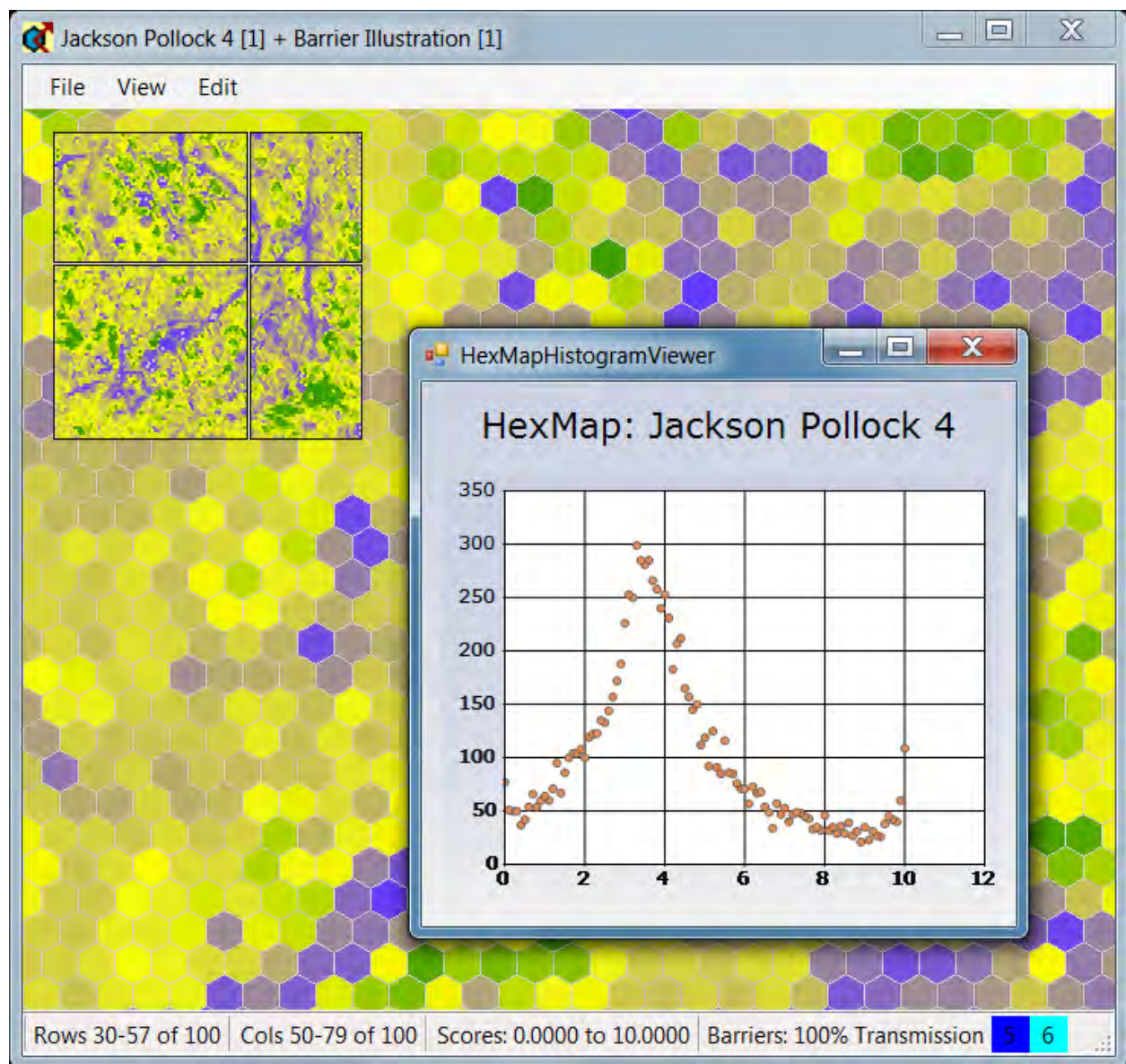
Barrier maps are collected into time series, just as HexMaps are. Each time step is a separate Barrier map, and each Barrier map can have its own collection of category pairs. In addition, each category pair's attributes are specified independently for each Barrier map. This means that barrier attributes can easily be made to change with time. This feature might be used to capture the increase in traffic on roads over the period of a simulation.

Multiple barrier time series can be used in a single simulation. For example, what constitutes a barrier might vary with life stage. This could be simulated by having different life stages respond to separate barrier time series.

Histograms

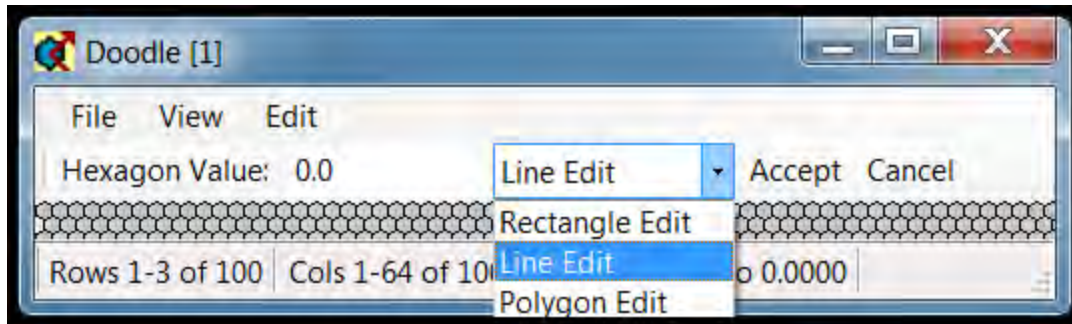
HexSim User's Guide

The File menu has an entry titled Show Histogram. This will create a simple plot showing the frequency of the hexagon scores found in the current HexMap. An illustration is provided below.



Editing Hexagons

The hexagon editing tool is available from the Edit menu. It is simple, but very useful. The HexMap being edited will be permanently altered once the Accept button is pushed. So users may want to edit copies of HexMaps, rather than the originals. Users simply select Edit Hexagons, and the Hexagon Editing Tool Bar will come up (see below).



The Hexagon Value field specifies what new score will be assigned to target hexagons. Users select between Rectangle, Line, and Polygon editing methods. These three methods are described below:

- **Rectangle Edit**

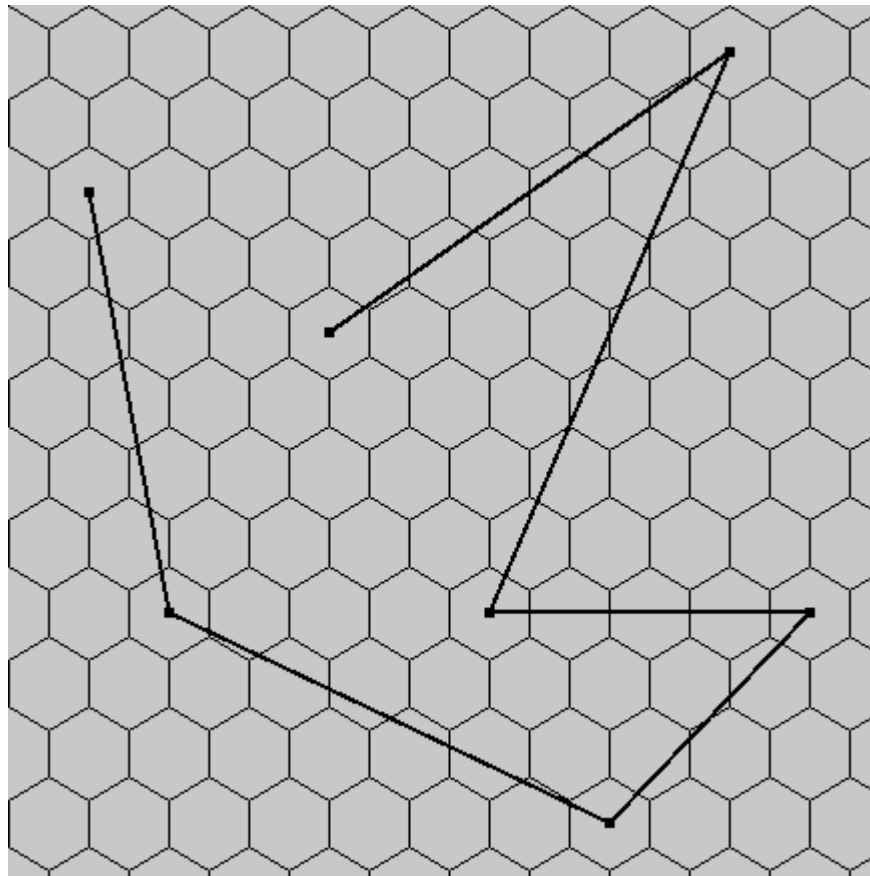
Click and drag the mouse. A rectangle will be created, and when the mouse button is released, the inscribed hexagons will be assigned the new score. Users may also click on a single hexagon to change just its value.

- **Line Edit**

Click and drag the mouse. A line will be created, and when the mouse button is released, the intersected hexagons will be assigned the new score. Users may also click on a single hexagon to change just its value.

- **Polygon Edit**

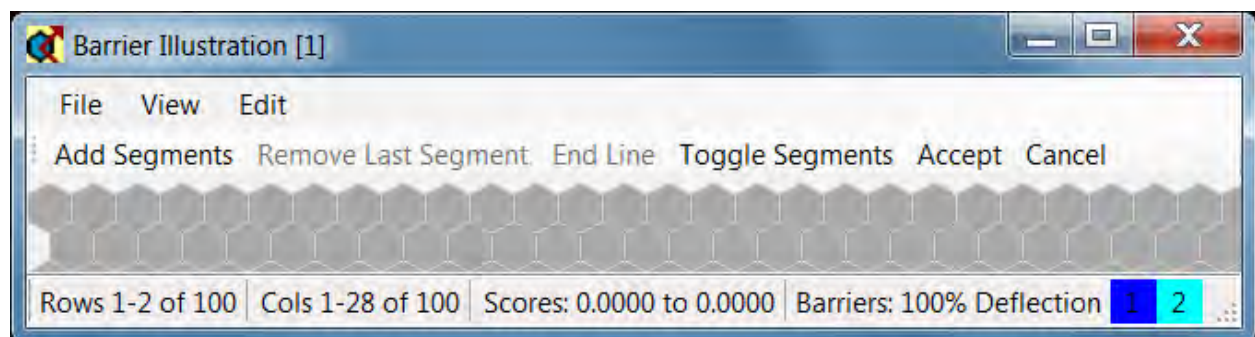
Click the mouse on multiple distinct hexagons. Each click will place a polygon vertex (see figure below). The vertices will always be located at the center of a hexagon. Use the Delete or Backspace keys to remove vertices. When the polygon is closed, the inscribed hexagons will be assigned the new score.



Editing Barriers

The first step in editing barriers is to select a barrier pair to edit. This step can be skipped if only one barrier pair has been created. Otherwise, the barrier pair to be edited can be selected using the Edit menu item labeled Set Current Barrier Category Pair.

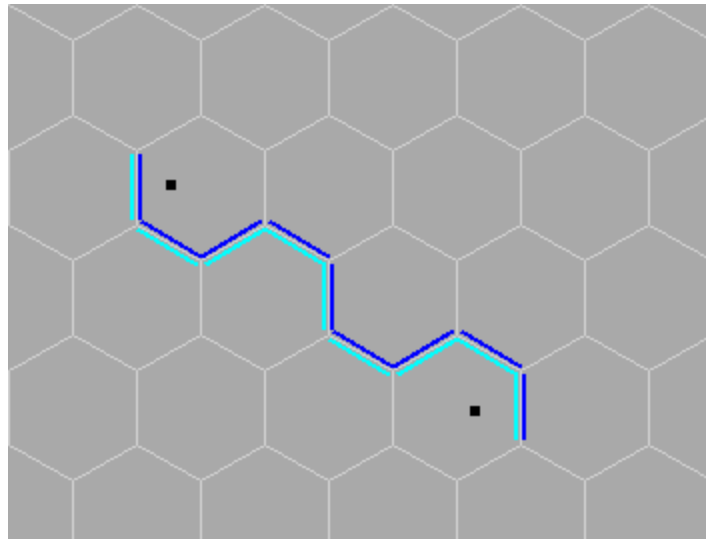
Barriers can be edited one edge pair at a time, or line segment by line segment. A segment is just a collection of edge pairs. When Edit Barriers is selected from the Edit menu, the Barrier Editing Tool Bar will come up (see below).



The separate buttons in the Barrier Editing Tool Bar are discussed below:

- **Add Segments**

Click the mouse at multiple points in the grid. Each click will place a polygon vertex, and every pair of vertices will be connected by a series of barrier pairs (see figure below). Unlike Polygon Edit with HexMaps, the vertices here are not fixed at the hexagon centers. The exact placement of a vertex can influence the line segment that is formed. Vertices can be removed using the Backspace or Delete keys.



Adding barrier segments with precise shapes can take some practice. To obtain a slightly different sequence of edges, try moving the mouse a slight bit, hitting the Backspace or Delete key, and then clicking again. This will effectively move a single vertex by a small amount, thus altering the sequence of edges that is constructed.

Barriers are typically added while a HexMap is being displayed (unlike the figures above). That way, the barrier edges can be placed on or around specific features in the landscape.

- **Remove Last Segment**

While barrier segments are being added, the last vertex can be removed using this button. This is equivalent to using the Backspace or Delete keys. Removing a segment is not equivalent to an undo operation. If segment Z is created such that it crosses segment X, then segment X is likely to be changed at the point of intersection. Removing segment Z will not undo the changes that were made to segment X. Instead, gaps will be left in segment Z where X had crossed it. For this reason, it is useful to accept your changes frequently. This will save the current editing session to the disk. Then if a problem arises, the current editing session can be cancelled with minimal loss of data.

- **End Line**

This will end the current line segment. The Spatial Data Viewer will remain in editing mode, so another segment can be added immediately after pressing this button.

- **Toggle Segments**

This will toggle barrier pairs on or off for a specific hexagon edge. The edge to be toggled is selected by clicking the mouse at the desired location. The barrier pair will be added with the blue edge closest to the mouse. Therefore toggling segments can be used to flip the orientation of the blue and cyan edges.

- **Accept**

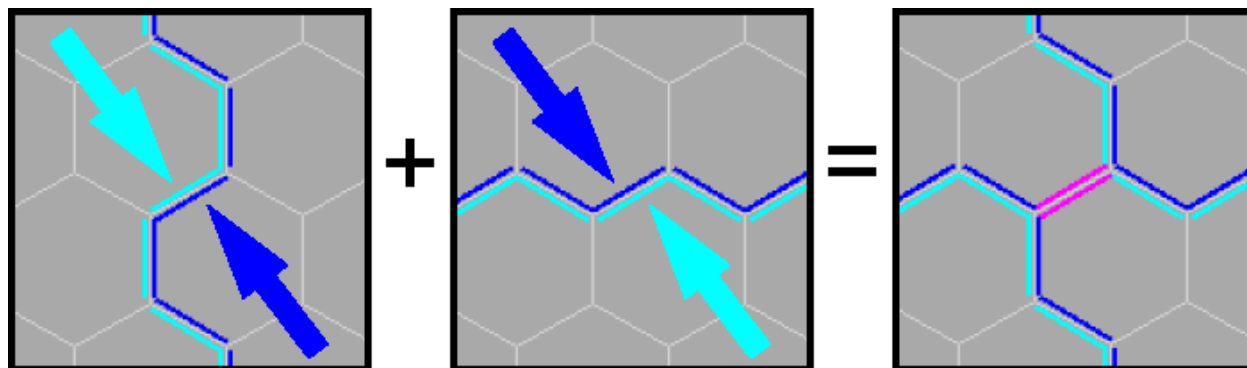
This will end the editing session and save any changes that were made.

- **Cancel**

This will end the editing session without saving any changes.

Barrier Conflicts

Conflicts can arise at edges when two barriers intersect. This will happen if the intersecting edges are associated with different barrier pairs, or different sides of the same barrier pair (see image below). HexSim displays these ambiguous edges in magenta so that users can easily locate them. The Toggle Segments barrier editing tool can be used to correct such conflicts. In the actual barrier file, conflicts are recorded by assigning the barrier pair ID a negative value. So in some cases, users may be able to resolve conflicts by directly modifying the barrier file in a text editor.



Barrier Maintenance Tools

Several tools are available for barrier maintenance. These are described below.

- **Edit Barrier Attributes**

This item includes a sub-menu that lets users select a specific barrier pair. Then the Edit Barrier Attributes window (shown below) is called up. The Edit Barrier Attributes window is where users supply mortality, deflection, and transmission probabilities for each category of a barrier pair. For each category, the three probability terms must sum to exactly 1.0. A Category Label can also be set. The Category Label can be used to better identify category pairs, when multiples exist.

Edit Barrier Attributes

Category Label: 100% Deflection

Category 1

Probability of Mortality: 0.0000

Probability of Deflection: 1.0000

Probability of Transmission: 0.0000

Category 2

Probability of Mortality: 0.0000

Probability of Deflection: 1.0000

Probability of Transmission: 0.0000

Clicking Accept will PERMANENTLY ALTER these Barrier Attributes BOTH IN MEMORY AND ON DISK.

Accept Cancel

- **Add Barrier Category Pair**

When this option is selected, a new barrier category pair is created, and the Edit Barrier Attributes window (shown above) is called up. The new category pair is loaded into the edit window.

- **Delete Barrier Category Pair**

This item includes a sub-menu that lets users select a specific barrier pair. If the barrier pair contains no edge data, it will be immediately deleted. If edge data has been added, a confirmation window will come up, and the user will have to dismiss this window before the deletion is processed.

- **Set Current Barrier Category Pair**

This item contains a sub-menu that lists all of the barrier category pairs. When one of these is selected, it becomes the current, or active barrier category pair. The

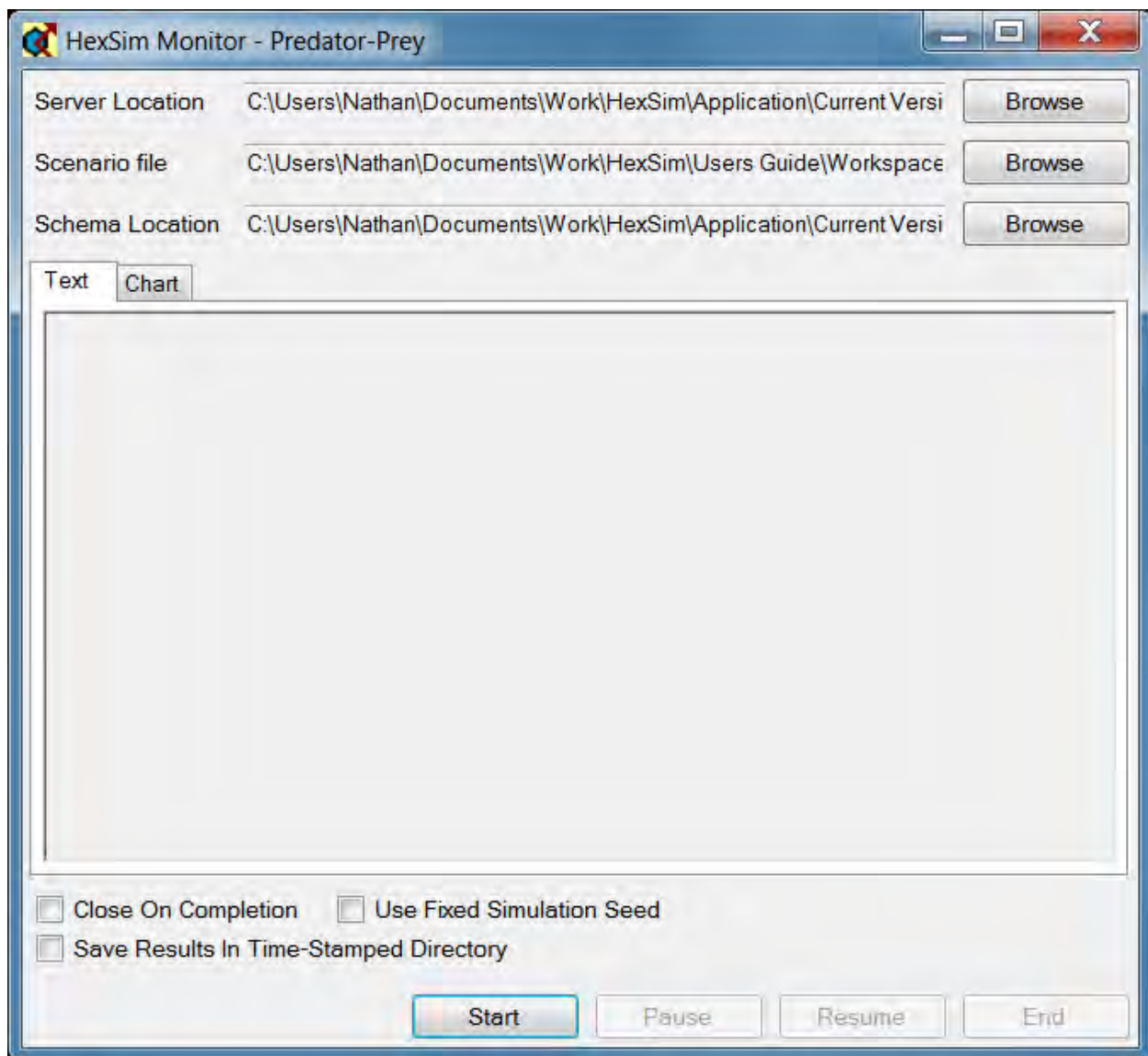
active pair is shown in blue and cyan. All other pairs are shown in black and white. Only the active pair can be edited.

- **Compact Categories**

Through combinations of adding and deleting barrier category pairs, it can happen that gaps arise in the numeric sequence. For example, pairs [1,2] , [7,8] , [15,16] might be present, and none other. The Compact Categories menu item can be used to collapse a sparse list like this down into a set of sequential pairs. If it was used in this example, then [7,8] would change to [3,4] and [15,16] would change to [5,6].

Running Simulations

HexSim is really a framework within which models can be developed and run. The models, in this sense, are scenarios. A simulation is the execution of a scenario. Simulations are started from the HexSim Monitor. The Monitor window is called up from the Scenario menu, by selecting the Run Simulation option. An image of the HexSim Monitor is shown below.



When called from the Scenario menu, the HexSim Monitor comes up with a server, scenario, and schema pre-loaded. If users launch the HexSim Monitor directly from Windows, then these fields will have to be populated by hand. The Server Location is

the path to the folder containing the HexSim application. The Scenario File is the path to the scenario being run. The Schema Location is the path to the scenario.xsd file currently being used. This file will typically reside in the HexSim application directory.

There are three check-boxes that give users additional control over a simulation. These are only active prior to starting a simulation. They cannot be changed while a simulation is running. The Close On Completion check-box will close the Monitor window as soon as the simulation has finished. The Use Fixed Simulation Seed check-box will force the random number generator seed to a fixed value. Using the fixed seed makes it possible for two users to exactly replicate each other's work.

The Save Results in Time-Stamped Directory check-box makes it convenient for users to modify and run a scenario over multiple iterations without overwriting prior results. Normally, if a user ran a simulation using a scenario called Test, the simulation results would be placed in the Results folder, in a sub-folder named Test. Running the Test simulation a second time would overwrite any previously stored results.

If the Save Results In Time-Stamped Directory is checked, then instead of creating a results folder named Test, HexSim would use a folder named Test_YYYYMMDDHHQQSSZZZZ, where

- YYYY is the year,
- MM is the month,
- DD is the day,
- HH is the hour, in 24-hour format,
- QQ is the minute,
- SS is the second, and
- ZZZZ denotes a sub-second temporal metric.

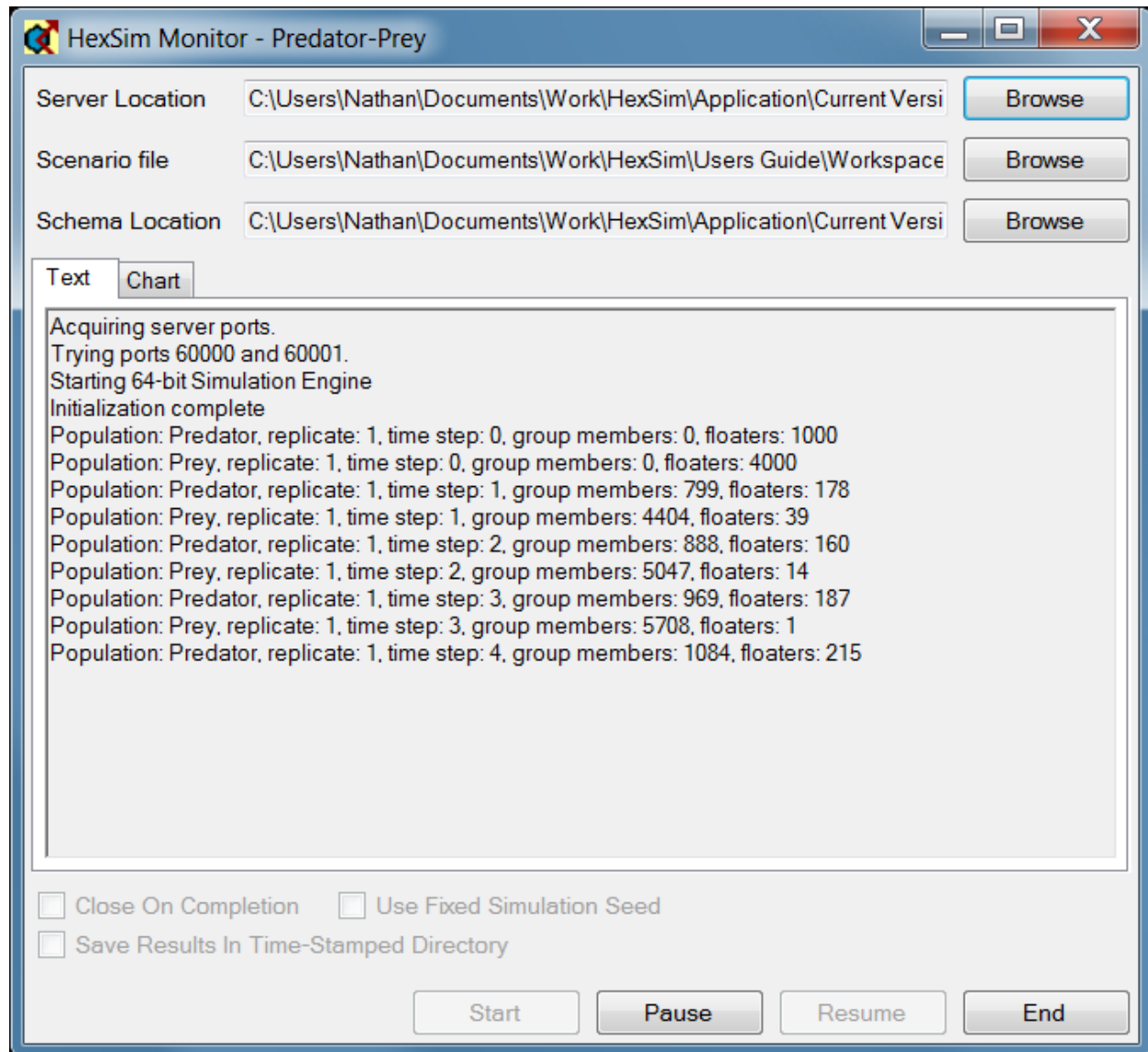
This scheme guarantees each results folder a unique name. For simplicity, the files within time-stamped results folders do not include the time stamp in their names.

When the Start button is pushed, the simulation will begin. The HexSim Monitor communicates to the model engine using inter-process communication through ports. So the first thing HexSim must do when starting a simulation is to find a pair of free ports it can use. These will typically be ports 60000 and 60001. But if these are being used by another application, HexSim will search for others. The figure below illustrates the feedback provided in the monitor's Text tab, as a simulation begins.

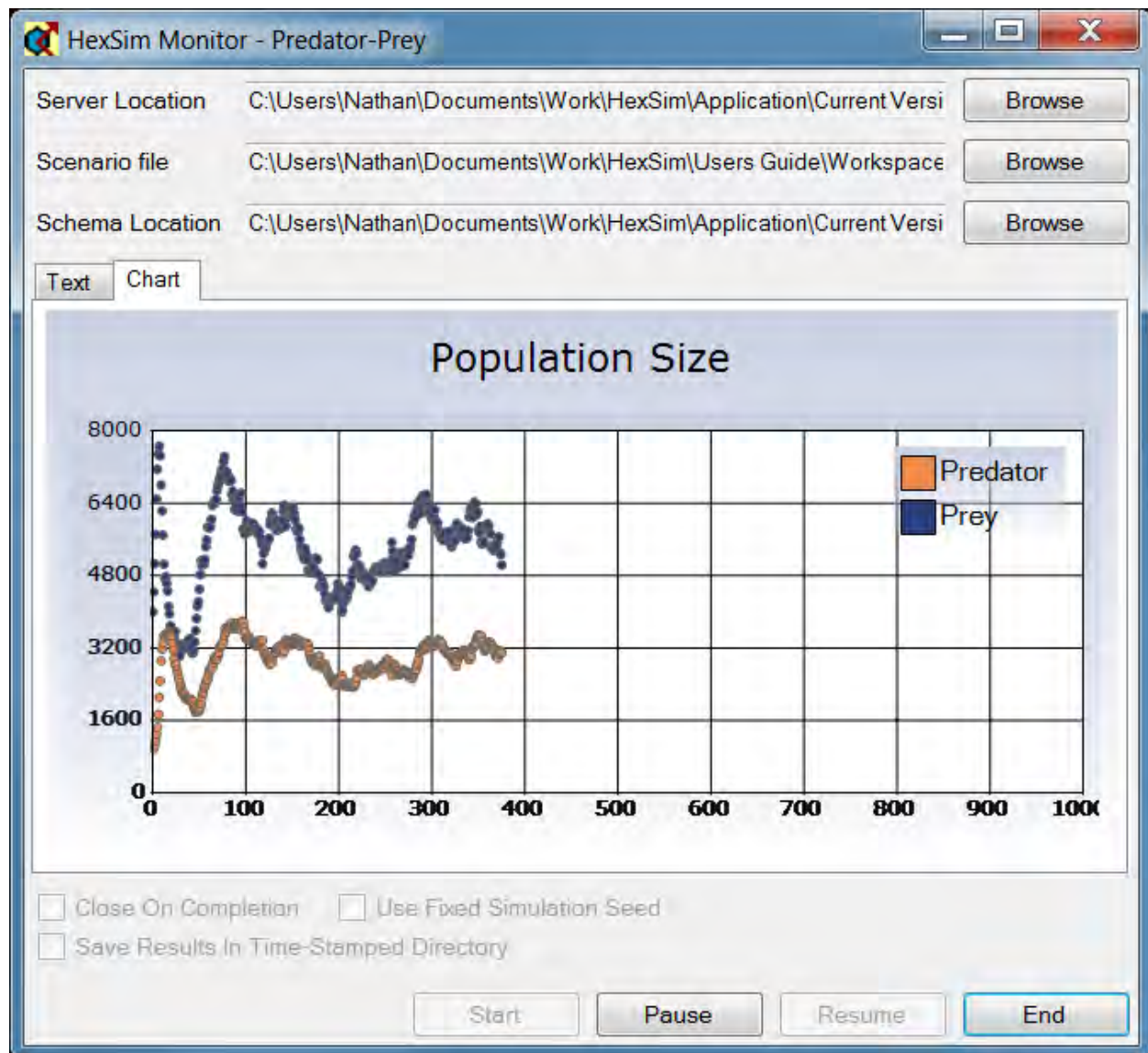
Once the simulation is running, the Pause and End buttons become active. If the Pause button is pushed, the simulation will immediately stop. Once paused, the simulation can be resumed or ended.

The End button terminates a simulation, but leaves the Monitor window up. Users may then edit and save a scenario, and immediately relaunch it by pushing the Start button

again. When this is done, the popup message warning users about results folder data loss will not be shown. This speeds up the process of launching successive simulations. But users must be careful to make sure they do not overwrite data they care about.



The Monitor window also includes a Chart tab. This tab shows a simple plot of each population's size through time. An image of the Monitor window's chart tab is provided below.



The HexSim Monitor window also includes an Errors tab. This tab is only displayed if errors crop up during a simulation. Errors are rare, since HexSim scenarios undergo a considerable amount of validation screening while they are being developed.

Population Parameters

Population Parameters

[The Properties Tab](#)

[The Range Tab](#)

[The Traits Tab](#)

[The Affinities Tab](#)

[HexSim Genetics](#)

[Resource Allocation](#)

The Population Parameters Properties Tab

Overview

An image of the Population Parameters Properties tab is shown below. The properties tab controls the initial population size and placement, whether an exclusion series is to be used, and whether genetics are implemented in the scenario.

Adaptive | Predator

Properties Range Data Traits Affinities Description

Terrestrial Population

Initial Population Size 1000

Initial Population Placement >>> Random <<<

Exclusion Series (only time step 1 will be used)

>>> None Selected <<<

☒ Exclude Zero-Valued Hexagons

☐ Exclude Hexagons Exceeding Zero

☒ Use Genetics

☐ Use Linkage

Recover Close

Initial Population Size

This parameter specifies the number of individuals that will be created at the start of each replicate. If multiple populations are being simulated, at least one initial population size must be non-zero. Other populations can be introduced subsequently using an Introduction event.

Initial Population Placement

Individual from each population are introduced when a simulation starts up, or as a result of introduction events. Users are given some control where these individuals are placed. Individuals can be located randomly, or based on HexMap quality. If initial locations are based on HexMap quality then the following steps are taken:

1. All the hexagons are put into a list that includes their ID and quality.
2. The list of hexagons is randomized
3. The list of hexagons is sorted by quality from best to worst
4. The first individual is placed in the first hexagon, the second individual is placed in the second hexagon, and so on...

Individuals are only placed into hexagons with a non-zero score. If the number of individuals exceeds the number of non-zero hexagons, then HexSim start over again at the beginning of the list of hexagons. That is, if there are N hexagons, then individual N+1 goes in the first hexagon in the list, individual N+2 goes into the second hexagon in the list, etc. If the HexMap selected has a maximum score of zero, then the population will be located randomly.

Exclusion Series

The use of an exclusion series makes it easy for users to keep members of a population from entering certain parts of a landscape. Exclusion series are ideal, for example, as a mechanism for keeping terrestrial organisms from walking over water bodies. For convenience, after selecting an exclusion series, users may then exclude either zero-valued or non-zero hexagons. Exclusion series are unique in that only time step 1 is used. This is done to eliminate the possibility that members of a population end up in excluded hexagons, which could happen if hexagons became excluded after a simulation had started. If movement restrictions must change through time, then this can be accomplished using dynamic HexMap or Barrier data.

Use Genetics

If this box is checked, then a new check-box titled Use Linkage is added. Also, a Loci panel is added to the Traits tab. The Use Linkage box instructs HexSim to use map distances when building child genomes from parent data. The Loci panel is used to define population genetics data. For more information, see [Populations / Genetics](#).

The Population Parameters Range Data Tab

Background

The range data are used in HexSim to control group and range formation. Groups are collections of individuals, such as packs, herds, flocks, etc. Ranges can be used to simulate defended territories, since they are non-overlapping. Home ranges might be approximated by explored areas, since these can extend beyond a range, and can overlap. The values in the range data tab are referred to as the "Static Range Parameters" because they are imposed at the start of every time step. The Adjust Range Parameters event can be used to alter these parameters, and any such modifications will remain in effect until either superceded by another Adjust Range Parameters event, or until replaced by the static range parameters at the start of the next time step. These values are used primarily by the exploration component of Movement events, and by Range Dynamics events, since these events principally control group and range construction, and maintenance.

And image of the Population Parameters Range Data tab is shown below.

Adaptive | Predator

Properties Range Data Traits Affinities Description

Range Spatial Data Fathom Five

Range Barriers >>> None Selected <<<

Maximum Range Area 85.0000 1.0000

Maximum Range Span 0.00000000 0.00000000

Maximum Group Members 2

Hexagons Range-Eligible if Value is At Least 0.0000

Minimum Range Resource 0.0000

Floater Preemption of Group Resources (%) 0.0000

☐ Competitive Ability (%) 0.0000

Resource Targets

☐ Capture Efficiency
☐ Has Mate
☐ Predation Level
☒ Sex
☐ Stages

Name	Rank	Target
Female	0	0
Male	0	0

Range Spatial Data

This field is used to select the spatial data that will be used to evaluate resource quality when ranges are constructed or modified. This is necessary because when groups are constructed or joined, an evaluation must be made of the available resources, relative to the groups resource demands.

Range Barriers

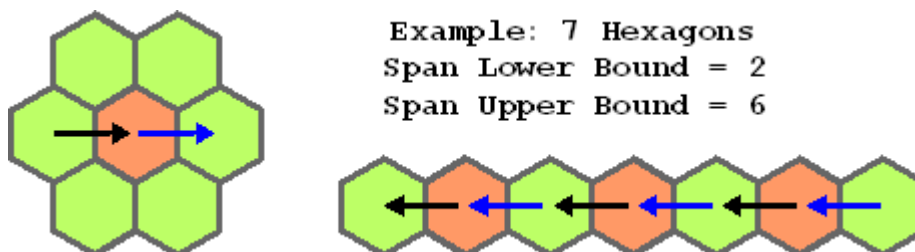
This field is used to specify which, if any, barrier data will be used in conjunction with range construction or maintenance. Ranges can wrap around the end of a range barrier, or penetrate gaps in barriers. But ranges will not be allowed to cross range barriers. In contrast to Movement events, the mortality, deflection, and transmission probabilities associated with barriers are ignored here. No consequences are imposed by range barriers other than the limitations they impose on shape and extent. Mortality and deflection probabilities can be imposed by assigning barriers to the Exploration component of a Movement event.

Range creation is controlled by Movement events, and barriers can be imposed in these events. However, ranges can be modified by Range Dynamics events, and when immigrants are accepted into a group. These range modifications are not limited by the barriers that may have been imposed on Movement. Thus even though users have established barriers to Dispersal and Exploration (both in Movement events), they may still observe ranges moving across these barriers as a result of range modifications. This barrier crossing can be stopped by specifying the barrier data in the range data tab.

Range Area and Span

These values specify the maximum allowed range size and span, in either hectares and meters, or hexagons. This data is stored in hectares and meters, so fractional values for hexagons are allowed. Range span is equal to the largest distance across the range, measured from centroid to centroid. For any given range size there is a minimum and maximum possible span. The maximum range span is always one less than the range area in hexagons (see figure below). The minimum range span is obtained when the range is constructed from concentric rings, since this is the most compact formation possible. For example, ranges of 2 or 3 hexagons can have a minimum span as low as 1 hexagon. But ranges of 4 through 7 hexagons have a minimum span that exceeds 1 hexagon (see figure below).

Since computing the minimum range span can be tricky, HexSim provides a context menu tool that set the span to the smallest even value that will not constrain groups from achieving the maximum range area. Depending on the value for maximum range area, the true lower bound on range span may be one less than this value. With the minimum and maximum span in mind, users can set the maximum range span parameter to a value that constrains range shape (lower values) or leaves it flexible (higher values).



The table below shows the number of hexagons in a compact cluster for the even values of span falling between 2 and 100. Compact cluster refers to a group built up from a central hexagon plus $[\text{span} \div 2]$ concentric rings.

Span	Size	Span	Size	Span	Size	Span	Size	Span	Size
2	7	22	397	42	1387	62	2977	82	5167
4	19	24	469	44	1519	64	3169	84	5419
6	37	26	547	46	1657	66	3367	86	5677
8	61	28	631	48	1801	68	3571	88	5941
10	91	30	721	50	1951	70	3781	90	6211
12	127	32	817	52	2107	72	3997	92	6487
14	169	34	919	54	2269	74	4219	94	6769
16	217	36	1027	56	2437	76	4447	96	7057
18	271	38	1141	58	2611	78	4681	98	7351
20	331	40	1261	60	2791	80	4921	100	7651

Maximum Group Members

This parameter sets the maximum allowed group size. Reproduction may cause group size to exceed this maximum, because new recruits are initially assigned group member status (in the natal group). Groups can also expand through immigration, but immigration is curtailed when the maximum group size has been reached. Floater Creation events convert group members to floaters, and can be used to bring group size back to the maximum.

Range Eligibility

Only hexagons with a score equal to, or exceeding this value will be eligible for inclusion in ranges.

Minimum Range Resources

This parameters sets the minimum cumulative value that all ranges must have. This value is the sum of the scores of all hexagons contained within the range. If this minimum value cannot be achieved, then the range will not be constructed.

Floater Preemption

This parameter specifies the extent to which floaters are allowed to take resources from groups. Floater preemption is useful when groups are allocated most of the resources in some portion of a landscape. In such cases, floaters may be unable to persist without some mechanism that provides them access to these group resources.

Both floaters and group members have explored and allocated areas. Group members' allocated areas are simply their ranges. Floater allocated areas consist of a subset of their explored areas not already allocated to other floaters. Hexagons within a floater explored area may be unclaimed, claimed by another floater, or claimed by a group. Floaters may be allocated all of the resources within unclaimed hexagons. Floaters may not obtain any resources from hexagons already claimed by another floater (this simplification was made to simplify and speed up the algorithm). Floaters may claim a portion of a hexagon that has been allocated to a group. The percentage of a range hexagon's resources that can be claimed by a floater is equal to the Floater Preemption of Group Resources parameter. Only one floater may preempt any given range hexagon.

When floaters build an allocated area from an explored area, they work from the hexagon last explored to the hexagon first explored. Each hexagon is examined to see if it is owned by another floater (in which case it is ignored), or if its owned by a group. If the hexagon is unowned, then the floater claims it and all of its resources. If the hexagon is owned by a group (and not preempted by another floater), and if preemption is non-zero, then the floater will claim access to the percent of its resources that can be preempted. This process continues until the entire explored area has been exhausted, or until the floater's target resource need has been met. The fact that a floater may have claimed a hexagon does not preclude a group from later adding it to its range. If a group claims a hexagon that forms part of a floater allocated area, the floater allocation will be reduced.

Any resources preempted by floaters are then no longer available to group members. So through preemption, floaters can lower group members productivity. Floater preemption can be easily turned off by setting this parameter to zero.

Competitive Ability

The Competitive Ability field is used to define the way that resources will be divided across individuals from different populations who occupy overlapping locations in space. Competition in this sense, only occurs between group members -- competition is not tracked between floaters, or floaters and group members.

The Competitive Ability updater acts on a single population, and it makes the following computations:

- A. Multiply each range hexagon score by a competition coefficient (see equation below).
- B. Sum up the post-competition resources available across the range.
- C. Divide the total available resource up among group members, taking resource rank (see below) into account.
- D. Compute the percent of each individual's target resource that this represents.
- E. Store this percentage in the individual accumulators.

The hexagon-specific competition coefficient used in step (A) is computed as follows. Assume there are k populations competing, let C_i be the competitive ability of the i^{th} population, and let $N_{i,x}$ represent the number of individuals of population i in hexagon x . Population i will be assigned a competition coefficient for hexagon x equal to

$$\frac{C_i \cdot N_{i, x}}{\sum_k C_k \cdot N_{k, x}}$$

This coefficient is then multiplied by the hexagon score to obtain the hexagon's available resource for the population under competition. The competition coefficient for any given hexagon is different for each population, but the sum for all competing populations is exactly 1.0. In fact, it is not even necessary that the population-specific Competitive Ability terms add up to 100%, although to do otherwise would needlessly obfuscate a scenario. For floater allocated areas, the competition coefficient is always 1.0.

The number $N_{i,x}$ above is recorded at the time that the Competitive Resources updater function is run.

If two or more populations compete for resources, then those populations should all use the same Range Spatial Data. To do otherwise would produce meaningless results.

Resource Targets

The Resource Targets parameters allow users to stratify individuals by trait, and for each trait value to supply a rank and resource target. This makes it possible to produce an unequal distribution of resources within groups. Each individual will attempt to attain their specified resource target, but the higher-ranked individuals will be served first. Thus if a group's resources fall below the sum of its members resource targets, some lower-ranked individuals will become resource-deficient.

A group's resources are equal to the sum of the scores of its range hexagons (computed using the current Range Spatial Data), minus any resources preempted by floaters. There are multiple ways that a group can become resource-deficient. These include resource scarcity, inter- and intra-group competition, floater preemption, landscape change, immigration, recruitment, individual trait changes resulting in increased resource demand, and Adjust Range Parameters events. Range Dynamics and Floater Creation events can be used to maintain parity between group resource demands and range resources. The consequence of range resource limitations is that it may cause the four acquired and competitive resource updater functions (within an Accumulate event) to return a value less than 100%, which can subsequently impact a resource acquisition trait. The resource acquisition trait may in turn impact many other events. Range resource limitations can also impact a group's ability to accept new immigrants.

Resource targets are unitless values that only have meaning relative to the range of scores present in the Range Spatial Data. However, the Range Spatial Data can be a time series, and it may also be changed using Adjust Range Parameters events. User's are encouraged to keep these details in mind and start with simple simulations that are easy to understand. Resource ranks indicate the order in which resources will be distributed. Higher rank individuals will have priority access to limited resources. Within a rank, individuals are sorted randomly before resources are allocated. Ranks may also be used to order individuals during the exploration phase of Movement events. In such cases, higher ranking individuals will get to explore first, and thus have greater access to limited resources.

Resource targets are used during the Exploration phase of Movement events, when starting new groups or joining existing groups. If the goal of Exploration is to start a new group, then exploration will continue until a range has been identified that can supply the individual's resource target, or until the maximum exploration area has been reached. Once the maximum exploration area has been reached, sub-standard ranges may be accepted. If the goal of Exploration is to join a group, then exploration will continue until a group has been identified that can supply the specified fraction of the individual's resource target. The receiving group is allowed to modify its range in order to acquire additional resources to provide to the immigrant. Resource targets are also used when Range Dynamics events expand, contract, or simply modify range structures. Resource targets are also used by Floater Creation events, and the four acquired and competitive resources updater functions (part of the Accumulate event).

Importing Range Data

Some HexSim simulations will use one or more Adjust Range Parameters events. The import button makes it easy to copy range parameters between the range data tab and Adjust Range Parameters events.

Non-territorial Species

Movement events only act upon floaters, and Reproduction events only act upon group members (the one exception being use of the Construct Explored Areas movement strategy, which operates on floaters and group members). So in most cases, simulations should involve both floaters and group members. It is straightforward, however, to convert individuals into floaters (use a Floater Creation event) for the purposes of movement. Then, movement can be used to convert the individuals right back to into group members. Importantly, HexSim's floater / group member paradigm does not necessitate invoking the notion of territoriality. Instead, HexSim's group and range machinery can simply be used to enforce population densities based on resource maps.

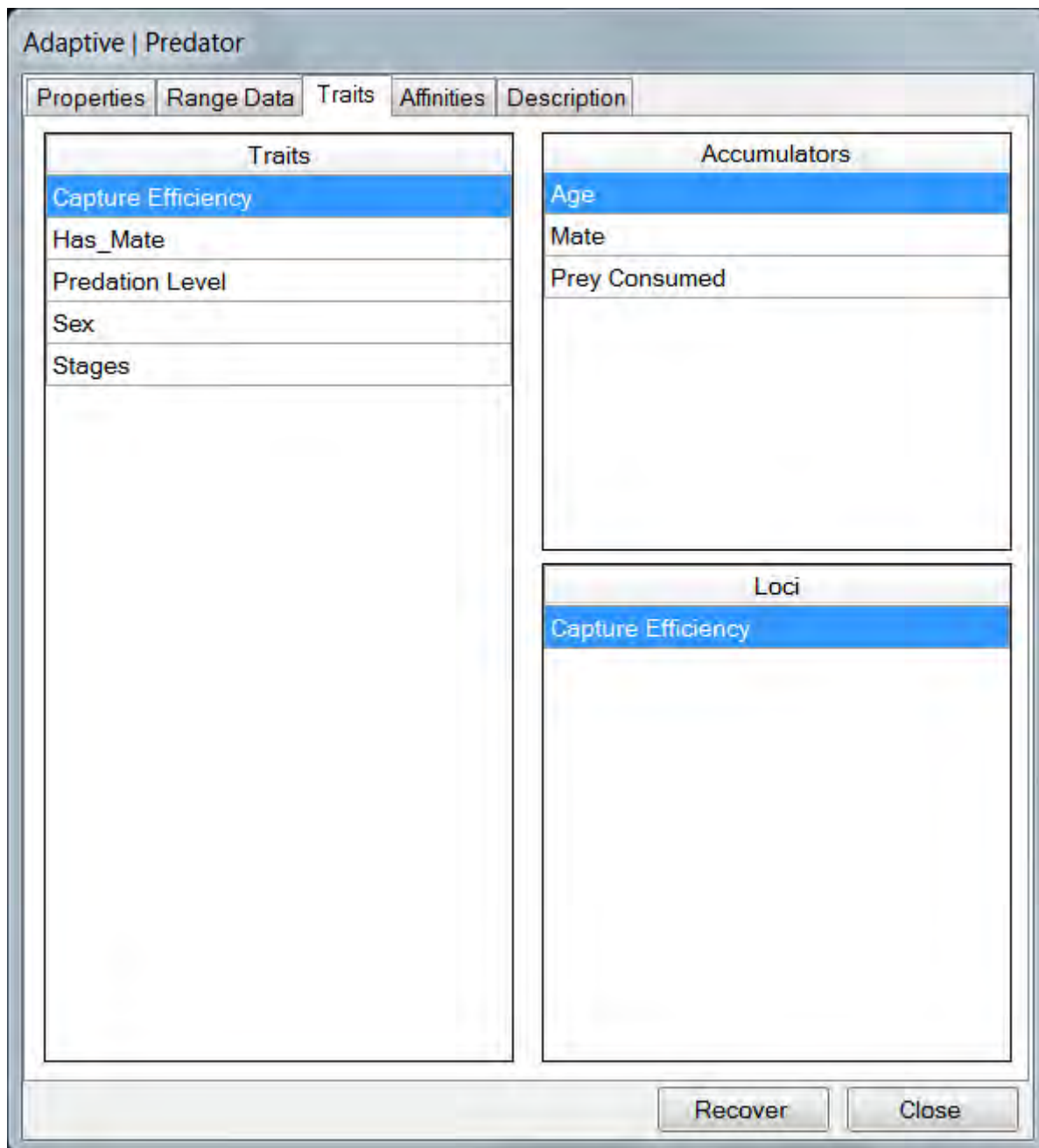
One approach to developing simulations for non-territorial species involves setting the target resource value to 1.0 for all selected trait values. Then a resource map can be developed for which the hexagon score is equal to the number of individuals a hexagon can support. If the target resource value is 1.0, then a hexagon with a score of 5.0 will be able to support five individuals. If an individual needs more than a single hexagon, then the hexagon scores should be less than 1.0. This resource map should then be used for the Range Spatial Data. The maximum group size should then be set no less than the maximum hexagon score. Individuals can be allowed to come and go from groups freely, and the range structure can simply be thought of as a way of imposing density limitations on the population.

The Population Parameters Traits Tab

Background

Traits are an essential part of HexSim. In fact, every population *must* have at least one trait. HexSim's traits can be probabilistic, accumulated (changing with time or exposure, etc.), or genetic. Traits are really just collections of mutually exclusive trait values. Individuals can only have a single trait value at a time, for any single trait. Part of a trait's definition is a set of rules governing the assignment of trait values. For example, a stage-class trait might be composed of three trait values: juvenile, subadult, and adult. This trait might then be set up so that the juvenile trait value is assigned to individuals in their first year of life, the subadult trait value is assigned to individuals in their second year of life, and all other individuals are assigned the adult trait value. Every trait must have at least two trait values. Every member of a given population will have the same traits, but they need not have the same trait values.

An image of the Population Parameters Traits tab is shown below. Accumulators are used in conjunction with accumulated traits. Accumulators are variables used to trigger changes between trait values (in the example above, the individual's age would be stored in an accumulator, and used to force transitions between stage classes). Probabilistic traits are parameterized with probabilities that control what trait value is assigned to each individual. Genetic trait values become a function of the alleles that are present at a given locus. The assignments are made using probabilities, but the right set of values can produce a deterministic outcome. Genetic traits can only be used when the Use Genetics check-box is selected, in the Properties tab of the Population Parameters window.



Accumulators

HexSim accumulators are simply variables. They are defined on a per-population basis, but separate copies are assigned to each individual. Accumulators can be used to track many different things, including age, exposure, mate ID, resource acquisition, etc. Accumulators are a necessary component of accumulated traits, and they have no other utility. Accumulators are altered by Updater Functions, which are accessed from within Accumulate and Interaction events. The accumulators panel includes a context menu with Add, Edit, and Delete functions.

The Accumulator Data window is invoked by double-clicking on an accumulator, or through the context menu. It is used to parameterize accumulators. This tool allows users to specify a name, minimum and maximum accumulator values, and initialization and birth values. Users are not obligated to impose a minimum or maximum accumulator value. Initialization and birth values are selected separately for each individual from a uniform distribution using the lower and upper bounds supplied by the user. If one or more parent traits are selected, then the upper and lower bounds used to establish the accumulator's value at birth will be stratified by the trait combinations. This would be useful, for example, if an accumulated measure of offspring fitness depended in part on parent fitness. It is perfectly OK to set the lower and upper bounds to the same value. Doing so just removes the stochasticity from that component of the process.

In the illustration below, parent traits are used to influence an accumulator tracking exposure. An exposure trait (not shown) is set to *exposed* when the exposure accumulator has a value of 1.0 or more. Otherwise the exposure trait is set to *unexposed*. In the illustration, infected and sick parents pass the disease to their offspring by forcing the children's exposure accumulator to be initialized to 1.0. This in turn influences the likelihood coming down with the disease (the probability of getting sick is only non-zero following exposure).

Accumulator Data

Name: Exposure

☐ Minimum

☐ Maximum

Parent Traits (Optional)

- ☒ Disease Status
- ☐ Disturbance Class
- ☐ Exposure Class
- ☐ Fitness Class
- ☐ Resource Class
- ☐ Stage Class

Accumulator Bounds

	Lower	Upper
Initialization	0	0
Healthy	0	0
Infected	1	1
► Sick	1	1

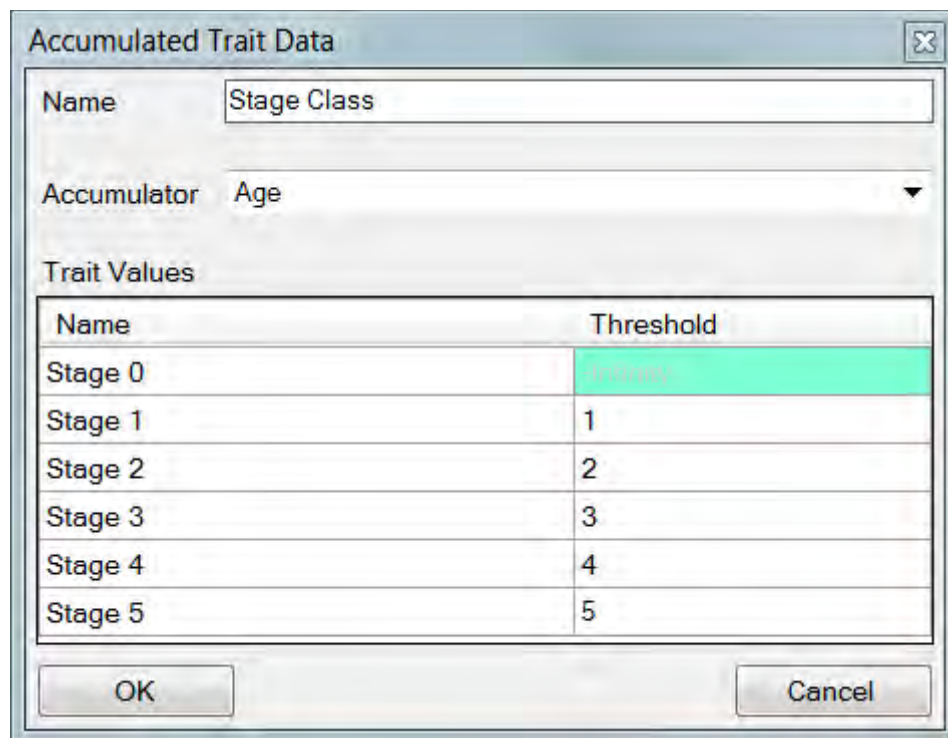
OK Cancel

Note that the four acquired and competitive resource updater functions are unique in that they always return a percentage value. They are used to store the percent of an individual's target resources that can be obtained from its range or floater allocation. No other updater functions return a percentage value.

Accumulated Traits

The traits panel includes a context menu with Add, Edit, Delete, and View Combinations functions. When accumulated traits are added or edited, an Accumulated Trait Data window opens. This tool allows users to specify a name, and accumulator, and a series of trait values. Trait values are added using the context menu. The trait values are paired with threshold values. When the referenced accumulator crosses one of the thresholds, then the trait value will change. The threshold values must be entered in increasing order.

Take for example an accumulated trait names Stage Class, linked to an accumulator named Age (see illustration below). Assume there are six stages labeled 0 through 5, and that individuals move sequentially through the stages each time step. If an individual's Age accumulator is equal to zero when it is born, then its Stage trait will be set to Stage 0 at birth. As soon as that individual's age increases to 1, the Stage trait will change to Stage 1, and so on. Individuals who reach stage class 5 will remain there for the rest of their lives.



Name	Threshold
Stage 0	-Infinity
Stage 1	1
Stage 2	2
Stage 3	3
Stage 4	4
Stage 5	5

The lowest threshold is unnecessary in this scheme, so this value is always fixed at negative infinity. In more complex cases, accumulators may increase or decrease in time, and they may do so in unequal increments.

Probabilistic Traits

When a Probabilistic Trait is added or edited, a Probabilistic Trait Data window opens. This tool allows users to specify a name, and a series of trait values. Trait values are added using the context menu. Each trait value must be paired with both initialization and birth probabilities, expressed as a integer percentage value. The probabilities specify the likelihood that each trait value will be imposed at initialization and at birth. Thus the initialization and birth probabilities must sum separately to 100%. Probabilistic trait values are subsequently modified using Transition events.

The example below illustrates how a probabilistic trait might be used to influence dispersal ability. Individuals would be randomly assigned to one of three dispersal classes at birth. Three different Movement events could then be used to make actual dispersal distance vary based on the Dispersal Ability trait value.

Name	Initialization (%)	Birth (%)
Short Distance	100	60
Medium Distance	0	30
Long Distance	0	10

Genetic Traits

HexSim's genetic traits take on values based upon the alleles present at a specific locus. So to use genetic traits, at least one locus must be defined. The Loci panel's context menu allows users to Add, Edit, or Delete individual locus definitions. By double-clicking a locus, or by using the context menu Edit option, the Locus Data window will be displayed. An example of this window is shown below.

Locus Data

Name: Capture Efficiency

Number of Alleles: 3

Inheritance Type: Both

Map Location: 0.0000

Initial Allele Frequencies:

☐ Use Regional Distribution

Values are: ☒ Identical ☐ User-defined

OK Cancel

In this case, the Capture Efficiency locus is assigned three alleles. And the initial allele frequency is set to $1/3$, $1/3$, $1/3$. The possible allele combinations any one individual can have are equal to the set of possible permutations, given that order is unimportant: (0,0) ; (0,1) ; (0,2) ; (1,1) ; (1,2) ; (2,2).

When a Genetic Trait is added or edited, a Genetic Trait Data window opens. This tool allows users to specify a name, a locus, and a series of trait values. Trait values are added using the context menu. Each trait value must be paired with a series of percentage likelihoods for each possible allele pair. Each column must sum to 100% to ensure that every individual is assigned one of the trait values, regardless of what alleles they possess. Thus genetic and probabilistic traits have similar parameterization schemes. The Capture Efficiency Trait Data window corresponding to the above example is shown below.

Genetic Trait Data

Name:

Locus:

Trait Values

Name	(0,0)	(0,1)	(0,2)	(1,1)	(1,2)	(2,2)
Low Efficiency	100	0	0	0	0	0
Medium-Low Efficiency	0	100	0	0	0	0
Medium Efficiency	0	0	100	100	0	0
Medium-High Efficiency	0	0	0	0	100	0
High Efficiency	0	0	0	0	0	100

OK Cancel

In this example, the probability values are specified so as to make capture efficiency vary with an individual's genome in a deterministic fashion. Transition events can be used to set the values of probabilistic traits based on the values of probabilistic, accumulated, or genetic traits. For example, a fitness trait might be varied based on the simultaneous influences of age, resource acquisition, genetics, and parasite load (which could be simulated probabilistically).

Genetic traits can also vary as a result of mutation. See [Events / Mutation](#) for more information.

Trait Builders

Traits are ubiquitous in HexSim simulations, and some specific trait types will be very common. So Trait Builders (wizards, in the Microsoft vernacular) were developed to automate the process of setting up a few useful or confusing trait types. At present, Trait Builders exist for Stage, Resource Acquisition, Floater/Group Status, and Stochastic Switch traits. These tools are dialog windows that prompt the user for requisite data.

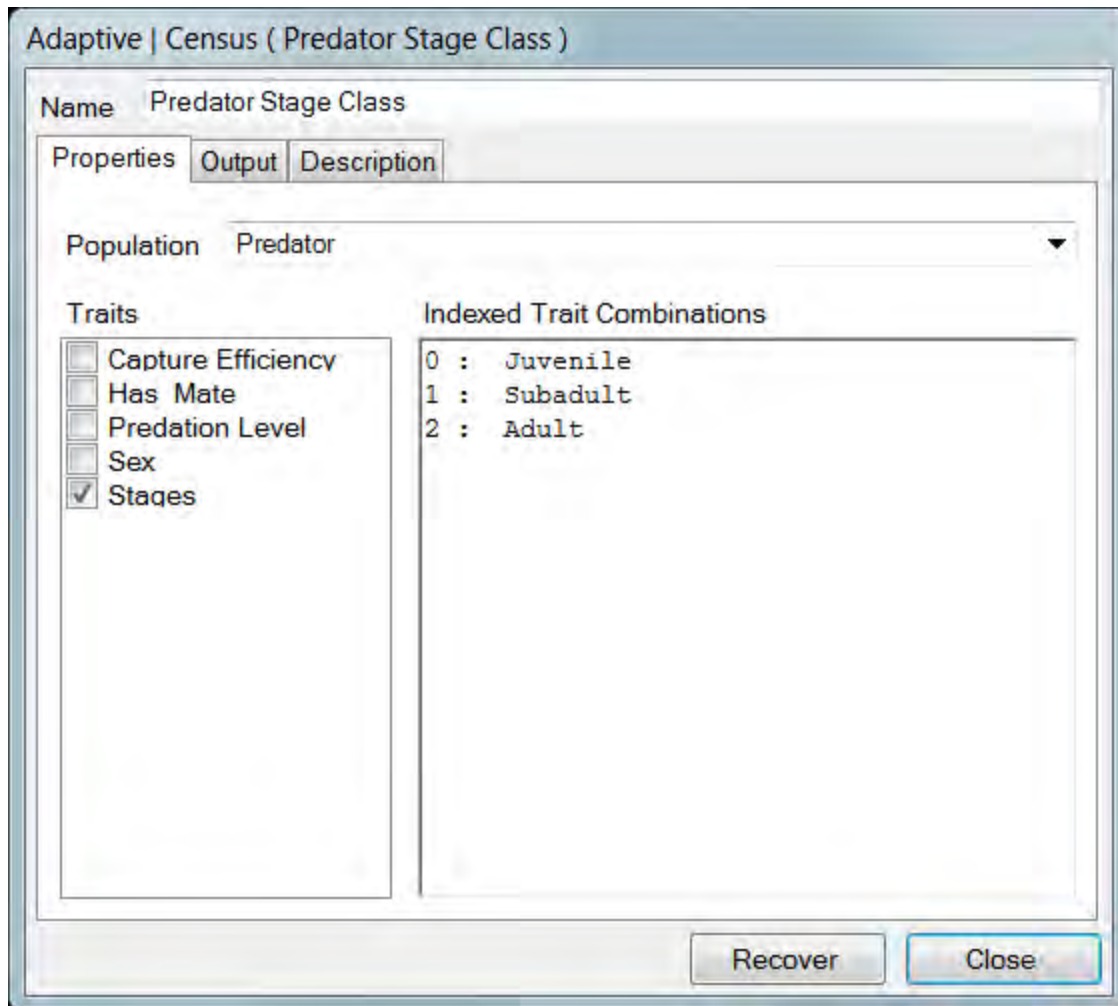
The Stage Trait Builder provides a choice of using either an accumulated or a probabilistic trait. When an accumulated trait is specified, the trait builder creates an accumulator, and it adds an Aging event to the life cycle. When a probabilistic trait is specified, the trait builder adds a Transition event to the life cycle. The Aging and Transition events will be colored gray as a reminder that they were automatically generated. The Trait Builder properties are discussed below:

- The Accumulated Resources Trait Builder creates an accumulator, and it adds an Accumulate event to the life cycle. The Accumulate event will be colored gray as a reminder that it was automatically generated. The Accumulate event will have a Resource Acquisition updater function. This updater returns a value equal to the percentage of an individual's target resources that can be obtained from its range or floater allocation.
- The Floater/Group Status Trait Builder creates an accumulator, and it adds an Accumulate event to the life cycle. The Accumulate event will be colored gray as a reminder that it was automatically generated. The trait is binary with Floater and Group Member trait values. The accumulator uses a Group Size updater function to capture the individual's group size. If the group size is one or more, the trait value is set to Group Member. If the group size is zero, then the trait value is set to Floater.
- The Stochastic Switch Trait Builder creates an accumulator, and it adds an Accumulate event to the life cycle. The Accumulate event will be colored gray as a reminder that it was automatically generated. The accumulator uses a Stochastic Trigger updater function to set the switch *Off* or *On* using the probability value supplied to the trait builder. The updater function returns a 1 with probability p (specified by the user) and a 0 with probability $1-p$. The accumulated trait switches from *Off* to *On* when the accumulator increases to one or more.
- The Stochastic Trigger updater function is unique in that it makes a single evaluation and then applies this result to every individual in the target population. Therefore, every individual in the population will have the same Stochastic Trigger trait value. A HexSim life cycle event can then be stratified by this trait, and applied only to individuals for whom the trait value is *On*. The result is that each time step the event will have a probability p of taking effect. So the utility of this Trait Builder is that the trait it creates can be used to make life cycle events fire probabilistically.

View Combinations

HexSim trait combinations are arranged internally with a fixed order. This is required so that users may change the names of traits and trait values. When HexSim writes census data to the disk, it uses this trait ordering to stratify population size by trait value. These output files are written in a comma-separated-variable (CSV) format, and they include a header. However, the headings for the trait-stratified population data columns are simply the internal trait index values. The View Combinations tool allows users to cross-walk these index values to the trait combinations they correspond to. The View Combinations tool has been integrated into Census events, and in most cases it is easiest to simply use that variant of the tool.

For example, the Census event depicted below will generate a CSV file. The right-most three columns of the file will be labeled Trait Index 0, Trait Index 1, and Trait Index 2. The first of these columns (Trait Index 0) will list the number of Juvenile individuals. The second column (Trait Index 1) will list the number of Subadults. The final column (Trait Index 2) will list the number of Adults. The View Combinations tool provides a way to access this mapping without creating a Census event.



Altering Trait Data

Removing a trait will invalidate any HexSim life cycle events that were stratified by that trait. In some cases fixing the event will involve simply selecting a replacement trait. But in other cases removing a trait will create more work for the user. For example, if a Survival event is stratified by a trait that is subsequently removed, then the survival rates entered into the event will be lost. This issue turns out to be fairly obvious once users become familiar with HexSim. However, adding or removing trait values (as opposed to deleting them can also lead to data loss in exactly the same way. This is a little less intuitive to new users of the model. Basically, changing the number of values a trait has can invalidate data tables associated with that trait. The most important thing

users must remember is to be careful when changing trait definitions, and always make sure a scenario is really OK before you save it.

The Population Parameters Affinities Tab

Background

Affinities are defined in the Population Data's Affinities tab. HexSim has four kinds of affinities:

- Natal Affinities
- Reproduction Affinities
- Resource Affinities
- Group Movement Affinities

Affinities are stored by Reproduction, Exploration, and Set Group Affinity events. Affinities are used by the dispersal component of Movement events. When affinities are supplied to dispersal, they control an individual's choice of direction through the autocorrelation mechanism. The result is that individuals tend to move towards the affinity site. If the affinity site is reached, then the dispersal process will terminate. Like traits, affinities are defined on a population-basis, but each individual holds their own unique copy of these data structures. An exception exists for Group Movement affinities, which are held by the group but inherited by individuals who leave the group.

An affinity data structure includes one or more records (hexagon IDs), and in some cases a threshold value. The thresholds are minimum values that must be met before an affinity record will be recorded. Thresholds are used for Reproduction and Resource affinities. Reproduction affinity threshold values are integers, while Resource affinity thresholds are real numbers. These affinity structures may store multiple records. Natal and Group Movement affinity structures are always limited to a single record in size. Reproduction affinity thresholds specify the minimum number of offspring that must be produced (by a single individual during a single Reproduction event) before a new record will be created. Resource affinity thresholds specify the smallest mean hexagon score that must be achieved before a new record will be created.

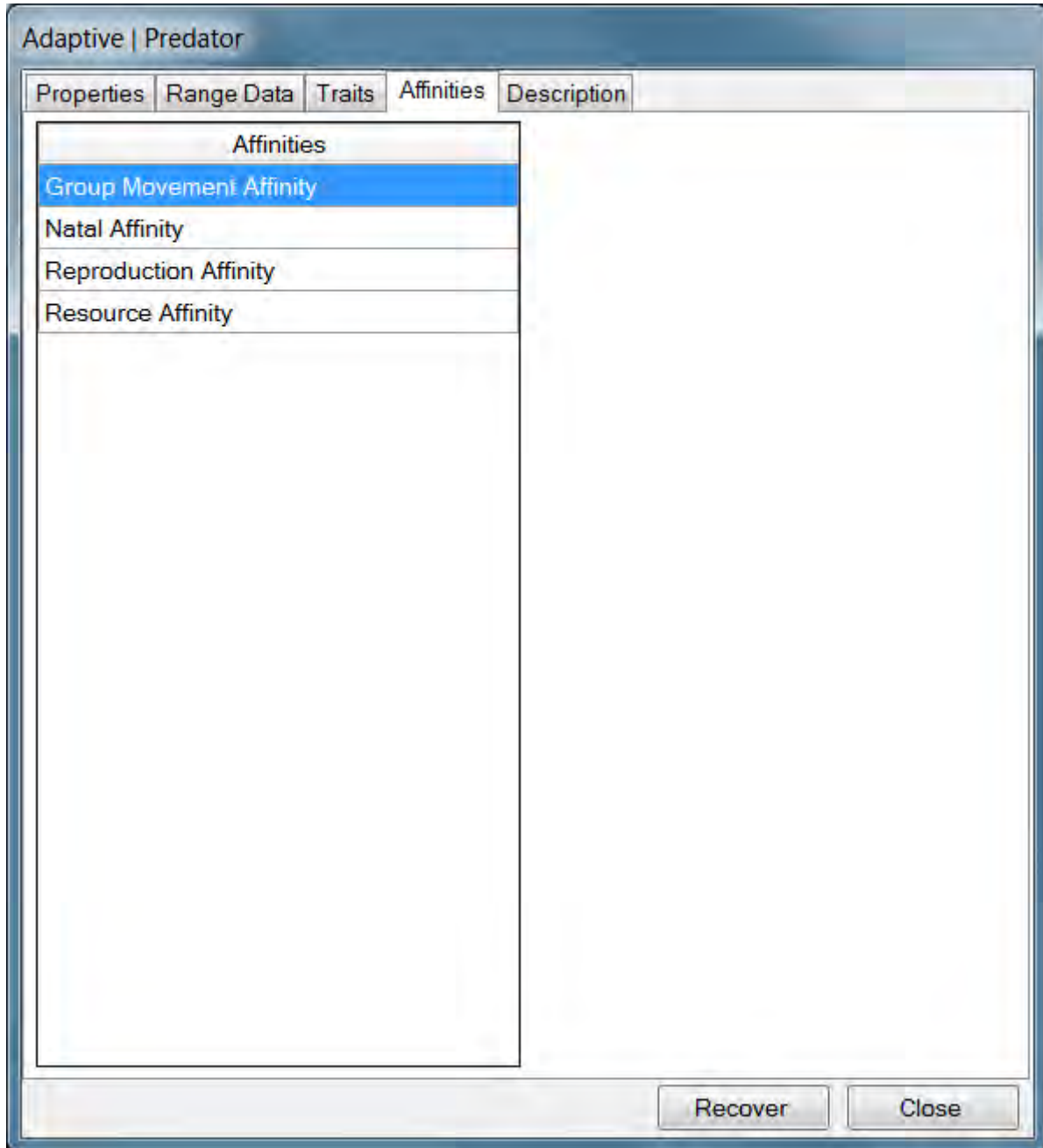
When the dispersal component of a Movement event is set to use an affinity, but multiple affinity sites have been stored, then one of the sites must be selected from the collection and used as the target. To accomplish this, a selection strategy is established when an affinity data structure is defined. Available strategies include selecting the Best, Closest, or a Random affinity site from those that have been stored. This selection process is only necessary when using Reproduction and Resource affinities.

If the selection strategy is set to Best, then the affinity records are sorted by quality, and the site with the best quality is selected. Ties are settled randomly. For Resource affinities, a record's quality is set equal to the mean hexagon score, taken over a range

or explored area. For reproduction affinities, a record's quality is equal to the reproductive output associated with that location.

If the selection strategy is set to Closest, then the record that is the shortest distance from the individual's present location will be selected. If the selection strategy is Random, then all records will have equal likelihood of being used., and the selection of a specific record is made at random. In both cases, ties are settled randomly.

An image of the Population Parameters Affinities tab is shown below.



Group Movement Affinities

Group movement affinities provide a mechanism for moving an entire group of individuals towards a common dispersal target. The Set Group Affinity event is used to identify an affinity site and assign it to all members of a group. If those group members subsequently become floaters and move, this group movement affinity can then be used to direct each individual towards the same target site.

Unlike the other affinities, Group Movement affinities are a property of the group itself. This allows them to be inherited by new members that join the group through immigration or recruitment. When a Floater Creation event forces individuals to leave a group, they make a copy of the group movement affinities, if any exist. From this point on, any group movement affinity records will be stored with the individual. If the individual subsequently joins a new group, then the group movement affinity data will be cleared.

This scheme is what makes group movement possible. The Set Group Affinity events assign an affinity to a group. Subsequent newcomers inherit this property. But for the group to move, it must first be broken up using Floater Creation. The floaters can no longer access the group properties, but they can use their local copies of the group affinity values. Thus each member of the former group will later be able to disperse to a common location.

A potential shortcoming may arise when a floater holding a group affinity fails to become a group member. The group affinity data will be retained as long as the individual remains a floater. This could cause it to keep attempting to disperse towards the affinity site, even though its former group may have long since moved on. On the other hand, this feature allows users to assign a group affinity, disband the group, scatter the former group members, and then reconstruct the group later using the still-valid affinity data.

Natal Affinities

Natal affinities record the birth hexagon. If the individual is born into a multi-hexagon range, the most central hexagon in the range is used as the birth hexagon. Natal affinities cannot be altered.

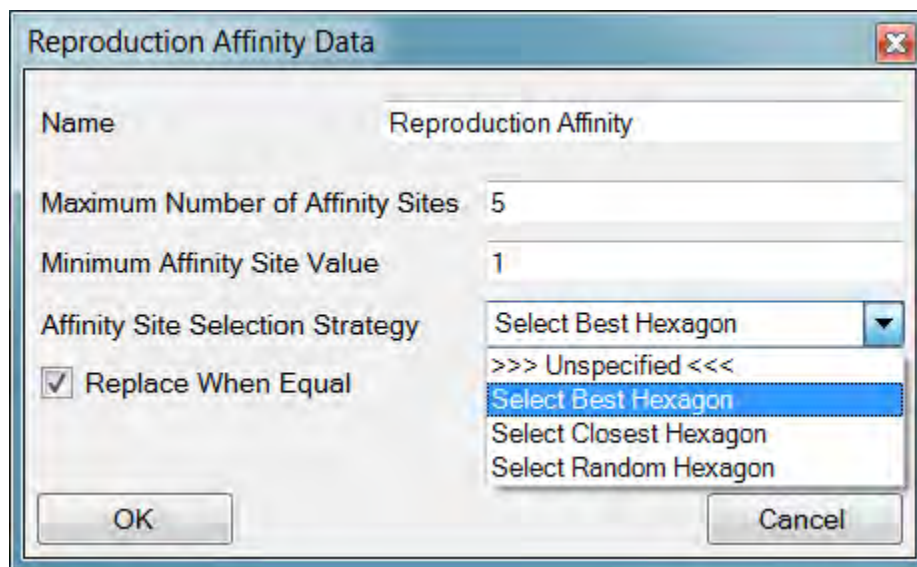
Reproduction Affinities

Reproduction affinities record sites at which reproductive output was high. Reproduction affinity records are stored by Reproduction events. Reproduction affinity records are collections of one or more pairs of numbers. Each pair consists of a location and a value. The location element is set to the most central hexagon in the individual's range. The location will be the same as the offspring's natal site. The value element is set equal to the reproductive output resulting from the Reproduction event that wrote the affinity record.

Data from a Reproduction event is only stored in an affinity record if the reproductive output (the number of offspring produced) meets or exceeds the minimum value specified in the affinity variable's definition.

A maximum number of records must be set when a reproduction affinity is defined. If the maximum number of records has been reached, then any subsequent affinity records will only be added if they can replace an existing record corresponding to a lower reproductive output. If the Replace When Equal check-box is unchecked, then new records will only replace old records that they exceed in value. If the Replace When Equal check-box is checked, then new records will replace old records that they meet or exceed in value. Collections of reproduction affinities may contain duplicate entries.

An image of the Reproduction Affinity Data dialog window is shown below.



Resource Affinities

Resource affinities record sites at which resource availability was high. Resource affinity records are stored by the Exploration component of Movement events. Resource affinities are only stored if the exploring individual (1) successfully starts a new group, (2) successfully joins an existing group, or (3) the exploration strategy has been set to Construct Explored Areas. Resource affinity records are collections of one or more pairs of numbers. Each pair consists of a location and a value. If the exploration culminates in a group being started or joined, the location element is set to the most central hexagon in the individual's range, and the value element is set equal to the mean quality of the range hexagons. If the exploration strategy is set to Construct Explored Areas, then the location element and value elements are set to the ID and score of the best hexagon in the resulting explored area.

The value stored in the affinity record is a mean hexagon score when ranges are started or joined. This value is a maximum hexagon score when ranges are being augmented

with new explored areas. Individuals that attempt to start or join a group, but fail, will not store a new affinity record. If the movement strategy is to Construct Explored Areas, both group members and floaters may store new resource affinity data.

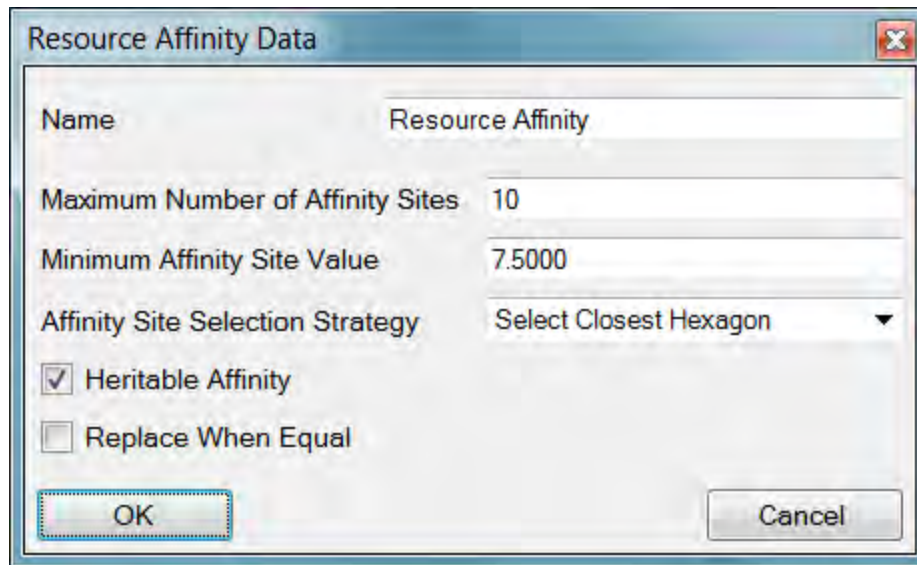
Data from a Movement event is only stored in an affinity record if the value element (the mean score of the range hexagons, or the maximum score present in the explored area hexagons) meets or exceeds the minimum value specified in the affinity variable's definition.

A maximum number of records must be set when a resource affinity is defined. If the maximum number of records has been reached, then any subsequent affinity records will only be added if they can replace an existing record corresponding to a lower value. If the Replace When Equal check-box is unchecked, then new records will only replace old records that they exceed in value. If the Replace When Equal check-box is checked, then new records will replace old records that they meet or exceed in value. Collections of resource affinities may contain duplicate entries.

If the Heritable Affinity check-box is checked, then children will inherit their mother's stored resource affinity data (just for the specified resource affinity). Otherwise, children will initially have no stored resource affinity data.

Resource affinities can become misleading if landscape change alters the value of hexagons that figured into the mean quality computation. A purging mechanism is provided to remove such stale affinity data. The purging option is available in the exploration component of Movement events, and is associated with the resource affinity selector. When purging is used, newly constructed explored areas are examined to see if they contain hexagons stored in any old affinity records. When such records are found, they are purged from the individual's resource affinity data structure. This keeps individuals from repeatedly coming back to a site that was good in the past but has since become degraded.

An image of the Resource Affinity Data dialog window is shown below.



The image shows a Windows-style dialog box titled "Resource Affinity Data". It contains several input fields and checkboxes. The "Name" field is labeled "Resource Affinity". The "Maximum Number of Affinity Sites" is set to "10". The "Minimum Affinity Site Value" is set to "7.5000". The "Affinity Site Selection Strategy" is set to "Select Closest Hexagon" with a dropdown arrow. There are two checkboxes: "Heritable Affinity" which is checked, and "Replace When Equal" which is unchecked. At the bottom are "OK" and "Cancel" buttons.

Field	Value
Name	Resource Affinity
Maximum Number of Affinity Sites	10
Minimum Affinity Site Value	7.5000
Affinity Site Selection Strategy	Select Closest Hexagon
Heritable Affinity	<input checked="" type="checkbox"/>
Replace When Equal	<input type="checkbox"/>

HexSim Genetics

Background

Genetics in HexSim consists of several interacting components. A genome is created for each population. Genotypes are assigned to the individuals that make up the starting population. Rules are established for inheritance. Mutation events may be developed and used to alter individual genotypes. And genetic traits must be developed that derive observable characteristics from individual genotypes.

HexSim genomes are a collection of loci, each having a user-defined set of alleles. HexSim alleles are represented as integers. HexSim permits both sexual and asexual reproduction to be simulated, and it allows users to specify that some genes are always passed from a particular parent. HexSim also provides a simple model for genetic linkage. Mutation is modeled using a HexSim event that can alter loci by changing one allele value to another. HexSim also includes a genetic trait type so allele pairs can be mapped onto trait values.

HexSim genetic can only be used in a scenario when the Use Genetics box has been checked. This check-box is located in the [Population Parameters' Properties tab](#). When the check-box is checked, a Loci panel is added to the Traits tab, just below the Accumulators panel. The Loci panel's context menu allows users to Add, Edit, and Delete locus data. After Loci data have been provided, users may use the Trait panel's context menu to add one or more genetic traits.

When genetics are used in HexSim, and when both male and females are being simulated, an option exists to simulate chromosome crossover, based on a map distance metric. This is implemented only if the Use Linkage check-box is checked. This check-box appears right below the box labeled Use Genetics (but only when Use Genetics is checked).

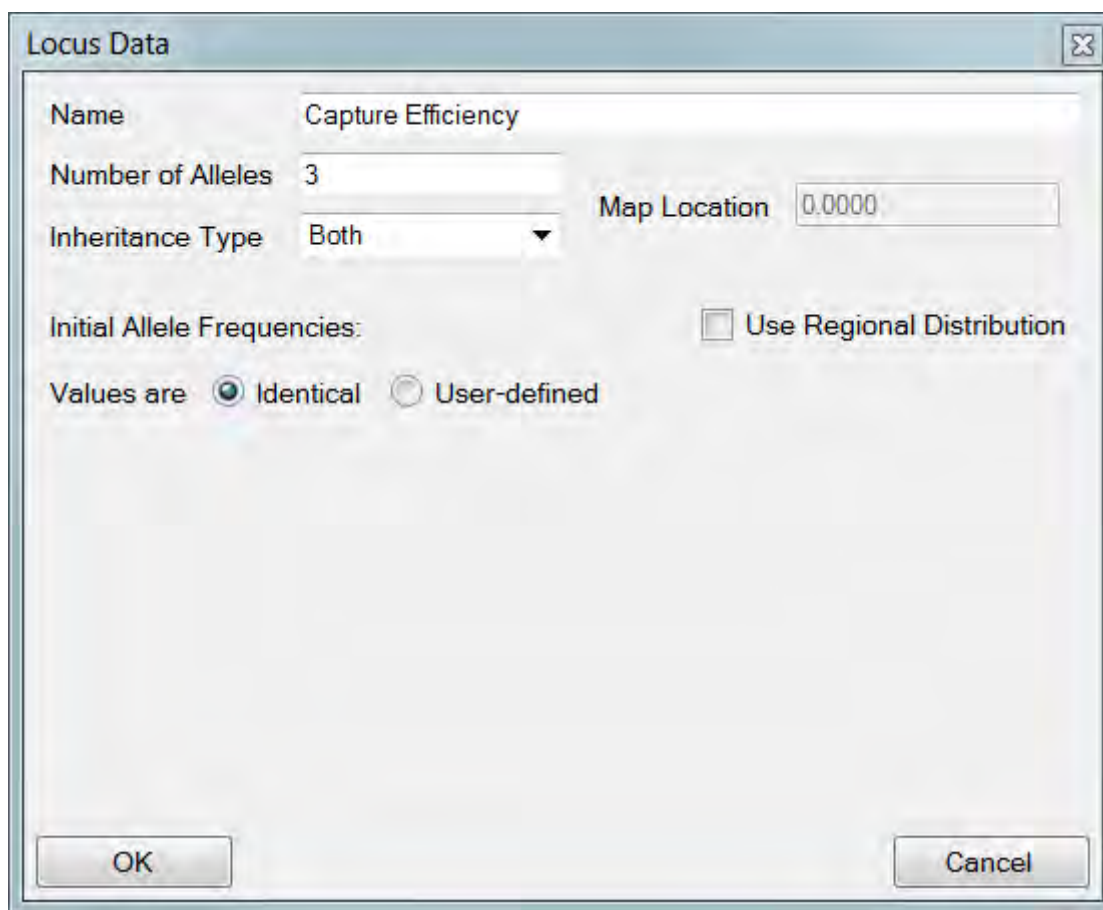
Loci and Alleles

The Loci panel's context menu allows users to Add, Edit, and Delete locus data. Each locus is assigned a name, a number of alleles, and an inheritance type. The inheritance type can be set to Mother, Father, or Both. The mother is the individual being acted upon by the Reproduction event, and the father is the mate (as specified in the Reproduction event). Users must be careful because this distinction between males and females is implicit. A clearly defined gender trait will help users avoid mistakes.

The inheritance type is irrelevant If the reproduction event ignores males. In this case, the offspring genomes will be exact copies made from the single parent. However, if

reproduction involves both a male and a female, and If the inheritance type is Both, then offspring genome will be assembled from a combination of both parents locus and allele data. Note that the inheritance type is specified on a per-locus basis. If there are specific loci that are always inherited from one parent, then the user can simulate this by setting the inheritance to Father (e.g. Y chromosome DNA) or Mother (e.g. Mitochondrial DNA).

An image of the Locus Data dialog is shown below.



The screenshot shows the 'Locus Data' dialog box. It contains the following fields and options:

- Name:** Capture Efficiency
- Number of Alleles:** 3
- Inheritance Type:** Both (selected from a dropdown menu)
- Map Location:** 0.0000
- Initial Allele Frequencies:**
 - ☐ Use Regional Distribution
 - Values are:** ☒ Identical ☐ User-defined

Buttons at the bottom: OK, Cancel.

The selection of alleles for any given locus is either random or non-random depending on whether map locations are specified or not. If the Use Linkage check-box has been checked, then loci must be assigned a Map Location. This value is a real number that only has meaning relative to the locations assigned to other loci. When the Use Linkage option is set, then loci will be sorted by map location. Otherwise they will be sorted in the order they were created.

Allele assignments are made at reproduction except for the initial population that is placed into the simulation at start-up. The allele frequencies of the initial population are specified in the Locus Data dialog window. If the Use Regional Distribution check box is not selected, then allele frequencies can be set identical, or specified individually by the user. No further input is required if the allele frequencies are made identical. When the

allele frequencies are "User-defined", each allele is assigned a percentage likelihood. The sum of the allele frequencies for all alleles must equal 100%.

When the Use Regional Distribution check-box is selected, then users must also pick a HexMap series to use in defining the regions. Only time step one of this series will ever be used for setting initial allele frequencies. When the initial allele frequencies are distributed regionally, a number of intervals must be specified. The range of hexagon scores is broken into the indicated number of intervals, and then users are allowed to set the allele frequencies separately for each interval. It is anticipated that a special HexMap will be developed exclusively for this purpose. Doing so will allow users to easily call out locations in the grid that are to be assigned unique allele distributions. Because this allele selection is part of the initialization process, each individual will be a floater located in a single hexagon. The single hexagon will have a score, and the score will fall in exactly one of the user-defined intervals.

In the simplistic example below, the HexMap "Matrix" has scores that range between 0 and 10. The locus was assigned three alleles, and the range of hexagon scores was broken into three equal intervals. Because the probabilities in this case are set binary (0, 100), individuals initialized into a hexagon of value [0, 3.33) will be assigned two copies of allele zero. Individuals initialized into a hexagon of value [3.33, 6.66) will be assigned two copies of allele 1. And individuals initialized into a hexagon of value [6.66, 10] will be assigned two copies of allele 2.

Locus Data

Name: Capture Efficiency

Number of Alleles: 3

Inheritance Type: Both

Map Location: 0.0000

Initial Allele Frequencies: ☒ Use Regional Distribution

Spatial Data Series: (only time step 1 will be used)

Matrix:

Number of Intervals: 3

Alleles ⇒ Intervals ↓	0	1	2
[0.00 to 3.33)	100	0	0
[3.33 to 6.67)	0	100	0
▶ [6.67 to 10.00]	0	0	100

OK Cancel

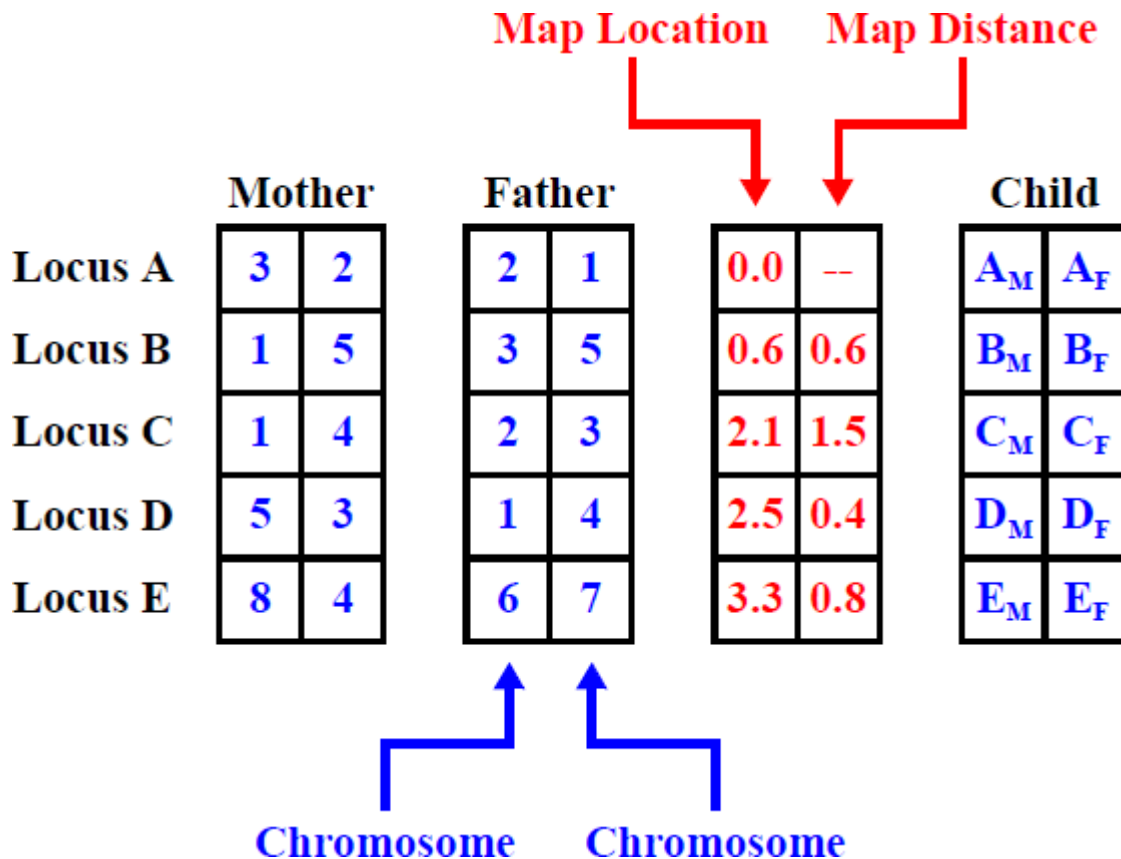
Reproduction

Users may configure either one or two-sex reproduction models in HexSim. In females-only reproduction scenarios, offspring genotypes are simply identical copies taken from the mother. In two-sex reproduction scenarios, males and females must first be paired, and then the Reproduction events can create offspring genotypes that recombine both parents genetic data. Users may implement females-only reproduction even if both males and females are present in a simulation.

The pairing of males and females can be performed directly within a Reproduction event. In this case, the pairing is strictly between members of the same group. These pairings will be temporary, lasting just for the duration of any one Reproduction event. These temporary pairings are created by checking the Use Mate Selection check-box in the Reproduction event. This adds a Mate Selection tab to the Reproduction event parameterization window. In the mate selection tab, users can specify a set of trait combinations that any mate must have to be selected. Only group members possessing the requisite trait combinations will reproduce, but having these trait values does not guarantee reproduction (for example, there may be more males in a group than females). Users may also specify whether mates will be selected with or without replacement.

Longer term pairings can be established through an Interaction event, with use of the Assign ID updater function. This updater function will assign a conspecific's ID to an accumulator, which can later be retrieved by a Reproduction event through the Mate Accumulator menu. Again, only paired individuals may reproduce. The Accumulate event can also be used to clear mate accumulators in case a member of a pair has died. This is accomplished using the Mate Verification updater function. When the assignment of mates is made through an Interaction event, it is then not necessary for reproduction that both members of the pair be part of the same group.

In HexSim, chromosomes are simulated as a vector of allele values. When Reproduction events perform recombination, the child receives two new chromosomes, one drawn from the mother's alleles, and one drawn from the father's alleles. These alleles may be selected at random from the two mother or father chromosomes, or with probabilities derived from map distances. Map distances are the separation between adjacent loci, as measured by their map locations (see figure below).



HexSim uses map distances to compute a probability that the next allele selected will be from the same chromosome as the previous allele. If map distance is 1.0 or more, an allele will be selected from the two mother or two father chromosomes at random. If Map Distance is less than 1.0, the probability that the next allele will be selected from the same chromosome as the previous allele is set to $1.0 - \text{map distance}$. A random number is drawn from $U(0, 1)$, and if the value is greater than $1.0 - \text{map distance}$, the allele is selected at random. Otherwise, the allele is drawn from the chromosome that

contributed the previous allele. The first allele (from the locus with the smallest map location value) is always selected at random. This allele selection process is performed independently for each parent. These simple rules are displayed schematically below.

$$\text{Map Distance} = \text{Map Location}_{i+1} - \text{Map Location}_i$$

If $\text{Map Distance} \geq 1.0$, allele selection is random.

else $P \equiv 1 - MD$.

If $x \in U(0, 1) < P$, allele selection is from the previous side.

else allele selection is random.

Through the careful use of map locations, users may control the likelihood of chromosome crossover. If crossover is of no concern, the Use Linkage check-box can simply be left unchecked. Then alleles will always be selected at random. Otherwise, certain alleles can be placed close to each other, thus lowering the chance of crossover, while others can be spaced farther apart. Any map distance of 1.0 or more guarantees a random allele choice. HexSim does not simulate multiple chromosomes. However, this can be effectively replicated by treating the multiple chromosomes as one much longer chromosome. If this is done, then the individual chromosomes should be separated by map distances of at least 1.0 (if crossover is being used).

Mutation

HexSim uses a Mutation event to simulate the process of genetic mutations. The Mutation event allows users to operate on one or more loci simultaneously. For a given locus, users supply probabilities that a mutation will switch one allele value to another. But these mutation rates can be stratified by probabilistic and accumulated (and genetic) traits. So, for example, it is easy to set up a Mutation event so that the likelihood of experiencing a mutation increases with exposure to some toxic substance.

Users must define the entire genotype in advance, so when mutations take place, they will typically involve "activating" alleles that were previously unused. Genetic traits can be developed that include values which are triggered when the mutated alleles are present, and that in turn alter behavior or fitness, etc.

The figure below shows a simple Mutation event parameterization. In this case, the event was stratified by a single gender trait, and the mutation probabilities were identical for both genders. There was no chance of experiencing a mutation that converted allele 0 to allele 2, or visa-versa. And there was a constant non-zero probability of experiencing a mutation from alleles 0 to 1, 1 to 0, 1 to 2, and 2 to 1. The white and green column coloring scheme is used to help users identify groups of transitions that have a common starting allele.

Adaptive | Mutation (Capture Efficiency)

Name Capture Efficiency

Properties Capture Efficiency Output Description

Allele Pairs⇒ Combinations ↓	0 to 1	0 to 2	1 to 0	1 to 2	2 to 0	2 to 1
▶ Female	0.001	0	0.001	0.001	0	0.001
Male	0.001	0	0.001	0.001	0	0.001

< 1 of 1 > Add Delete

Recover Close

See the section of this User's Guide on the [Mutation event](#) for further details.

Resource Allocation

HexSim populations are composed of individuals. Each individual has a location, and regions of space that it can access. These regions are referred to as explored and allocated areas. Explored areas are collections of hexagons that an individual has visited. Allocated areas are collections of hexagons that an individual has more exclusive access to.

Explored and allocated areas are used by several updater functions for resource acquisition, and to track exposure to landscape features. Interaction events also depend on explored and allocated areas. For organisms that defend a territory, HexSim ranges best approximate the defended territory, because ranges in HexSim do not overlap. HexSim explored areas best approximate a home range, because HexSim's explored areas can overlap. Of course, HexSim can also be used for organisms that do not defend a territory. Individuals can be collected into groups for reproduction, and broken out of groups for movement. Resource acquisition, the tracking of exposure, interactions, etc. can all be performed using explored areas, ranges, or floater allocations.

HexSim individuals are either group members or floaters. Group size can be one or more, but group members always have access to a set of hexagons called a range. Group members share the range resources, although this does not have to be done equitably. Group members also have explored areas, although when a group is formed or joined, its explored area is set equal to its range. For more information on range construction, see the section on the [Population Parameters Range Data Tab](#). A Movement event strategy called Construct Explored Areas can be used to build group member explored areas that extend beyond their ranges.

Group member locations, when needed for reports and tallies, are drawn at random from their ranges. Floater locations are always the last hexagon that the floater dispersed to. When HexSim first starts up, each member of the initial population has its floater locations set to the hexagon it was placed into.

Floaters are individuals without a group, and hence without a range. Floaters still have explored and allocated areas, but they differ from those of group members. Floater explored areas are the hexagons it has most recently dispersed through, or explored. Both dispersal and exploration begin by clearing any previously stored explored area. Floater allocated areas are a subset of their explored areas. To determine allocation to floaters, HexSim iterates over the explored hexagons in the reverse order of their exploration. Resources are allocated to a floater until its target is fulfilled or all explored hexagons have been processed. Floaters may add explored hexagons to their allocated area if the hexagons are not part of another floater's allocation. If a hexagon is part of a group's range, then floaters may only add it to their allocated area if the Floater

Preemption of Group Resources parameter is set greater than zero in the Population Parameters Range Data Tab. While floater allocations may overlap with group ranges, only a single floater may preempt any given range hexagon.

Unlike floater allocations, ranges are owned by an entire group. And ranges may not overlap. However, ranges can change constantly, so the ownership by groups of any one hexagon may not be at all static. Ranges are initially constructed by a single individual, and they are formed from that individual's explored area. This explored area is created during the exploration component of a Movement event. When individuals join a group through immigration, the group's range is allowed to expand to accommodate the newcomer's needs. This is not true when groups increase due to recruitment. Births and deaths can leave groups with ranges that are larger or smaller than they really need. The Range Dynamics event can be used to make corrections in these cases.

Explored and allocated areas are both cleared and created by Movement events. Explored areas are created during both the dispersal and exploration components of movement. But allocated areas (both ranges and floater allocations) are only created during exploration. If a Movement event consists of dispersal only, then each mover will end up being a floater with no allocated area. If a Movement event includes exploration, then every individual will develop a new explored area. Individuals that end their exploration as group members will be assigned a range. Those that end their exploration as floaters will instead be assigned a floater allocation. No other HexSim events modify explored areas, or floater allocations. Range Dynamics and Floater Creation events can alter ranges and group membership, but they do not change explored areas or floater allocations. When Range Dynamics modifies ranges, the group members' explored areas are left unchanged. When Floater Creation turns a group member into a floater, the individual retains its explored area, and is left with no floater allocation.

Movement only operates on floaters, unless the movement strategy is set to Reset Explored Areas. In that case, movement operates on both floaters and group members. When the exploration strategy is set to Reset Explored Areas, then floater explored and allocated areas are both updated. That is, a new floater explored area is constructed, and a new floater allocation is developed from that new explored area. When Reset Explored Areas applies to group members, their explored areas will be updated, but their ranges will not change. The range hexagons will be added to the new explored area.

When a HexSim simulation first starts, the initial population will not have explored or allocated areas. For that reason, a Movement event is automatically added when a population is created. This Movement event will trigger only at time step one, and it will assign each individual an explored and allocated area.

HexSim Events

HexSim Events

[Event Basics](#)

[Accumulate](#)

[Adjust Range Parameters](#)

[Census](#)

[Floater Creation](#)

[Generated HexMap](#)

[Interaction](#)

[Introduction](#)

[Movement](#)

- [Global Tab](#)
- [Dispersal Tab](#)
- [Exploration Tab](#)

[Mutation](#)

[Range Dynamics](#)


[Reproduction](#)


[Set Group Affinity](#)

[Survival](#)

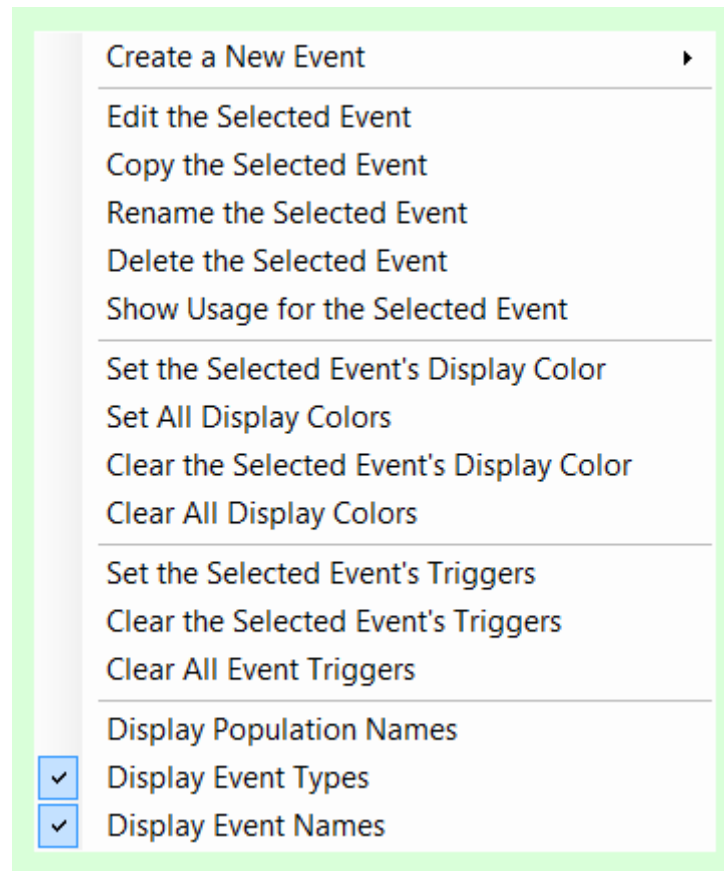
[Transition](#)

Event Basics

Each HexSim event has a unique purpose, interface, and parameter set. Two things that most events do have in common are Output and Description tabs. Output tabs are used to control what data the event writes to the simulation log file. Since log files can easily become quite large, there can be value in turning elements of the logging process off. Logging is always set on by default. If any part of logging is turned off, a  icon will be placed to the right of the event type, in the main window's Event Sequence. The description tabs allow users to record notes, which will then be stored in the scenario XML file.

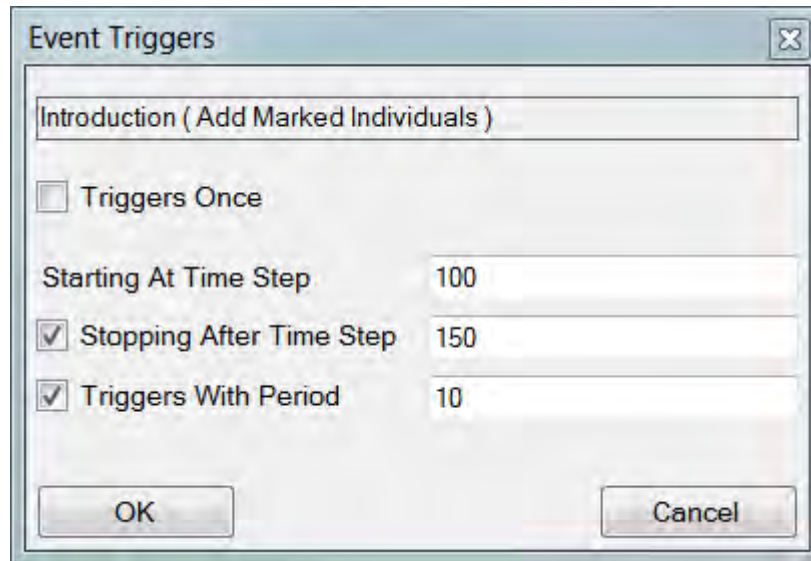
HexSim includes an extensive validation scheme that attempts to flag invalid parameter values. In most cases, invalid parameters and events will be identified with a  icon. Also, a tool-tip (a small message window) will pop up when the mouse is placed over the red exclamation point. These tool-tips can be valuable parameterization aids.

All of the basic operations that act on events can be accessed through the event context menu (see image below). This context menu has options for event creation, editing, copying, renaming, and deletion. The context menu also provides access to event coloring, triggering, and some display parameters.



Each HexSim event can be assigned a color. This is done using tools available in the event context menu. Coloring events can add clarity to the life cycle structure. Automatically generated events will be colored black (when added by population creation) or gray (when added by trait builders), but these colors can be changed.

Event triggers can be used to make events active for just a subset of the simulation time steps. Events may trigger once, or during a period defined by a starting and ending time step. In the latter case, a periodicity setting is also available. For example, suppose an event starts at time step 5 and has a period of 3. The event will then execute on time steps 5, 8, 11, 14, etc. An event that starts at time step 5 and has a period of 10 will execute on time steps 5, 15, 25, 35, etc. When triggering is set, an icon will be displayed in the event sequence, just to the left of the event. A → icon is displayed when the event will trigger just once. A ↗ icon indicates that the event will trigger more than once. The triggering conditions will be displayed in a tool-tip when the mouse is placed over a triggering icon. When triggering is off, events will be active in every time step. In addition to the context menu, the triggering dialog can be accessed by double-clicking the triggering icons. An image of the triggering dialog window is shown below.



All HexSim events operate on a single population except for the Interaction and Persistent Generated HexMap events. HexSim events can be used multiple times in a single simulation. The order of events in the event sequence reflects the order in which they will actually execute (except for any complications due to triggering).

As HexSim events run, they write data to the simulation log file. The log file is the data repository accessed by the model's reporting features. The Census and Persistent Generated HexMap events are exceptions in that they write their own results directly to the results folder. Census events record data on population trends in CSV files. If multiple census events are used, then the output CSV files will be assigned a numerical tag, with the number indicating the event's order in the event sequence. Persistent Generated HexMap events can be set to write HexMap files directly to the results folder.

The Accumulate Event

Background

The Accumulate event is really just a structure for housing updater functions. Updater functions assign values to accumulators. When an accumulator's value is modified, the traits linked to it are automatically updated. Therefore Accumulate events are the events that force accumulated trait values to change. HexSim provides a number of updater functions that perform very different tasks. More than one updater function may be used within the same accumulate event. Every updater function is supplied with a single accumulator name, and this determines which accumulated trait it acts upon.

When an Accumulate event is first created, a single "Global" tab is present in the interface. Updater functions that are placed in the global tab will operate on an entire population. If users right-click on the global tab, a context menu will appear that allows a "Stratified" updater tab to be created. Updater functions that are placed in stratified tabs only operate on the segment of a population having the selected trait combinations. Multiple stratified tabs may be added to the interface. Updater functions are not required in the Global tab if at least one has been added to a Stratified tab.

The order in which updater functions are called can make a difference in a simulation. Within the global, or any one stratified tab, the updater functions are called in sequence from top to bottom. Up and down arrows (the blue bars in the figure) are available for reordering the updater functions when multiple exist within a single tab. If one or more stratified tabs are used, then the tabs (and the updater functions they contain) are processed from left to right.

The remainder of this section is devoted to Updater Functions. An image of an Accumulate event is depicted below.

New Scenario | Accumulate (Multiple Updaters)

Name Multiple Updaters

Properties Description

Population Sneetches

Global Stratified Stratified Stratified

Updater	Target Accumulator
Increment	Age
Individual Locations	Location
Mate Verification	Mate

Recover Close

Accumulator Minimum and Maximum

When HexSim Accumulators are defined, minimum and maximum values may be specified (these are optional). However, only certain updater functions respect these bounding values. The rule of thumb is that updater functions that assign a value do not respect the accumulator minimum and maximum values. Updater functions that increment do respect the minimum and maximum values. Below is an exhaustive list of accumulators, grouped by event (Accumulate versus Interaction) and behavior.

Accumulate Events

Accumulator Minimum and Maximum Ignored

- Allocated Hexagons
- Births
- Clear

- Explored Hexagons
- Group Size
- Individual Locations
- Mate Verification
- Quantify Environment
- Resources (Acquired)
- Resources (Competitive Acquired)
- Resources (Explored)
- Resources (Competitive Explored)
- Stochastic Trigger
- Time Step

Accumulator Minimum and Maximum Respected

- Accumulator Transfer
- Increment
- Stochastic Increment

Interaction Events:

Accumulator Minimum and Maximum Ignored

- Clear
- Assign Value
- Assign ID
- Assign Accumulator Value

Accumulator Minimum and Maximum Respected

- Increment
- Increment From Accumulator

Temporal Smoothing

Some of HexSim's updater functions have a Temporal Smoothing parameter. This allows the accumulator value to represent an average, rather than just a single point in time. But true averages are costly to implement (in processing time and memory), especially given that population sizes might be quite large, and the average could be taken over many time steps. For that reason, HexSim uses a computationally efficient approximation to an average. To remind users that a true average is not being taken, we refer to this function as Temporal Smoothing instead of averaging.

Temporal Smoothing, as implemented in HexSim, can be described as follows: Let **OLD** refer to a previous (stored) accumulator value, and let **NEW** refer to a new value just measured. Further, let **N** represent the temporal window size (**N** is the parameter associated with smoothing). Then the accumulator will be set to a value **Z** given by:

$$Z = \frac{Old \times (N - 1) + New}{N}$$

Only the accumulator's present value need be stored to perform this computation, regardless of the size of the temporal smoothing window.

Accumulator Transfer

The screenshot shows a dialog box titled "Updater Function Data". It contains the following fields and controls:

- Function Type:** A dropdown menu set to "Accumulator Transfer".
- Target Accumulator:** A dropdown menu set to "My Accumulator".
- Target's Traits:** A text field containing "Categories".
- Source Accumulator:** A dropdown menu set to "Another Accumulator".
- Source's Traits:** An empty text field.
- Transfer Amount:** A text field containing the value "1".
- Buttons:** "OK" and "Cancel" buttons at the bottom.

The Accumulator Transfer updater function transfers value from one accumulator to another. Users specify both source and target accumulators, and then a transfer amount. To the extent possible, the transfer amount is then moved from the source accumulator to the target accumulator.

Accumulators may be assigned minimum and maximum values when they are defined (this can be changed at any time). If they are set, the Accumulator Transfer updater function respects the minimum and maximum values. Thus in some cases the transfer amount may be reduced in order to keep one or both accumulators within these limits.

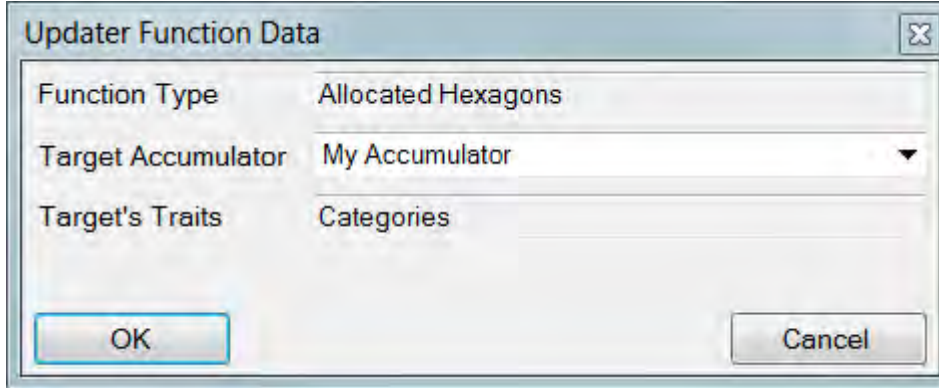
For example,

- Assume $x = 7$ is limited to $[0, 10]$, and Accumulator Transfer is told to shift 10 units from x to y . Only 7 units will be shifted, since transferring more would drive x below 0.
- Assume $y = 5$ is limited to $[0, 10]$ and Accumulator Transfer is told to shift 10 units from x to y . Only 5 units will be shifted, since transferring more would drive y above 10.

Users should be careful if using Accumulator Transfer with unbounded accumulators, since this can cause accumulator values to become arbitrarily large (positive or

negative). Similarly, when using Accumulator Transfer with bounded accumulators, users should keep in mind that the transfer amount may be truncated.

Allocated Hexagons



The dialog box titled "Updater Function Data" has a close button (X) in the top right corner. It contains three labeled fields: "Function Type" with the value "Allocated Hexagons", "Target Accumulator" with a dropdown menu showing "My Accumulator", and "Target's Traits" with the value "Categories". At the bottom, there are two buttons: "OK" and "Cancel".

The Allocated Hexagons updater function records the number of hexagons in a group member's range, or the number of hexagons in a floater's allocated area.

Births



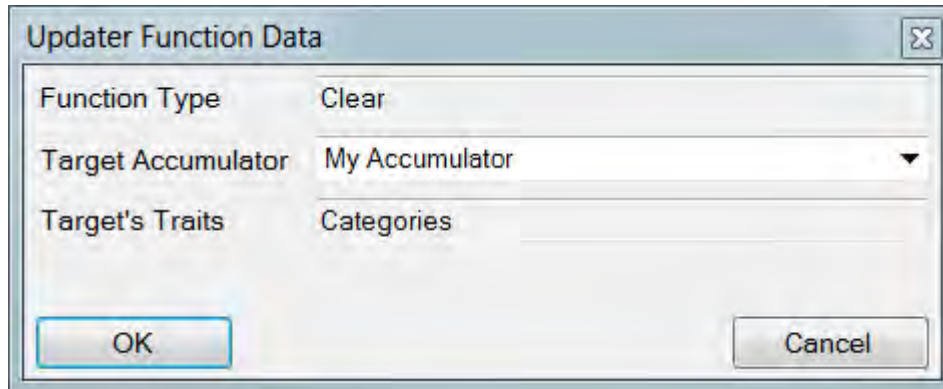
The dialog box titled "Updater Function Data" has a close button (X) in the top right corner. It contains three labeled fields: "Function Type" with the value "Births", "Target Accumulator" with a dropdown menu showing "My Accumulator", and "Target's Traits" with the value "Categories". Below these fields is a checkbox labeled "Compute Running Total" which is currently unchecked. At the bottom, there are two buttons: "OK" and "Cancel".

The Births updater function quantifies individual reproductive output. Each individual retains a record of the number of offspring it produced in the most recent reproduction event. The updater function reads this value and assigns it to the accumulator, but does so in one of two ways. If the Compute Running Total check-box is checked, then the accumulator will be set to the cumulative number of births. Otherwise, it the accumulator will be set to the actual number of births last recorded.

It is possible to use the Births updater function incorrectly. If the Compute Running Total check-box is checked, and the updater function is called more than once per reproduction event, then the target accumulator's value will exceed the actual

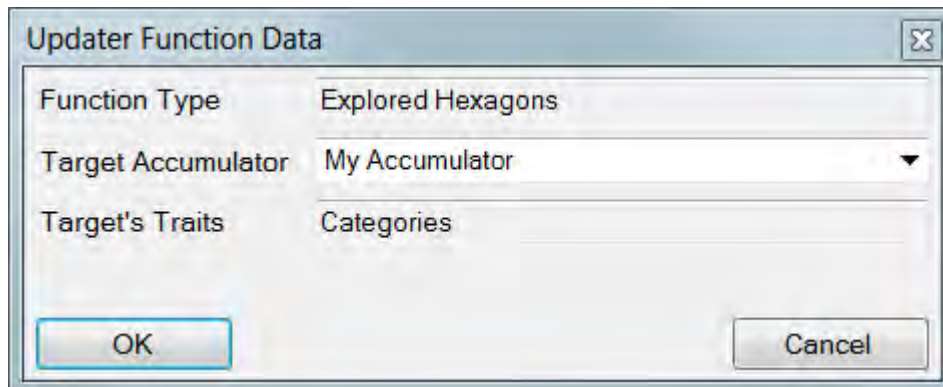
reproductive output. Similarly, if there are more reproduction events than there are accumulate events, the births accumulator will under-count the number of births.

Clear



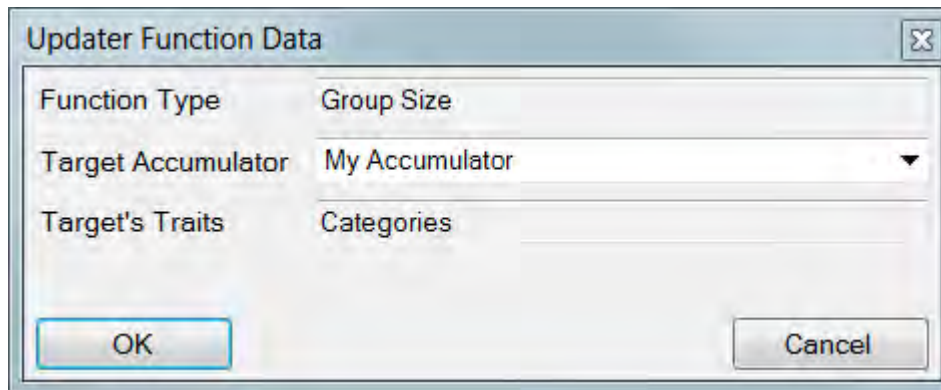
The Clear updater function simply sets an accumulator to zero.

Explored Hexagons



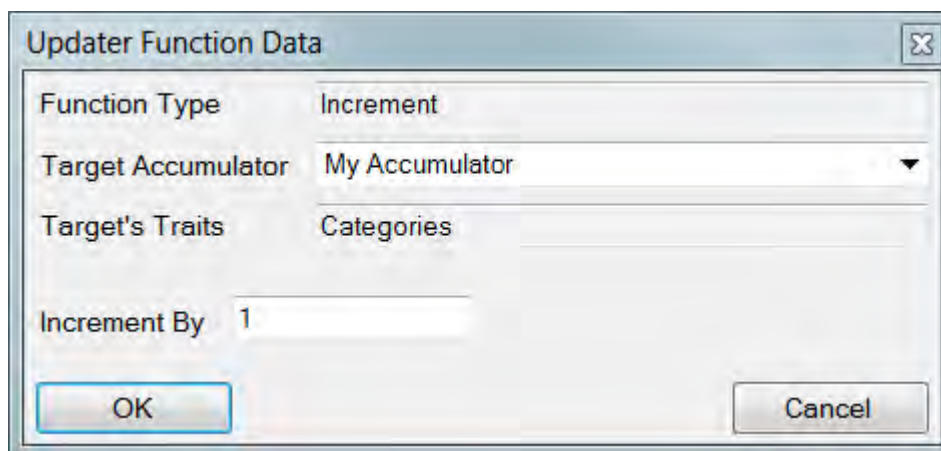
The Explored Hexagons updater function records the number of hexagons that comprise each individual's explored area.

Group Size



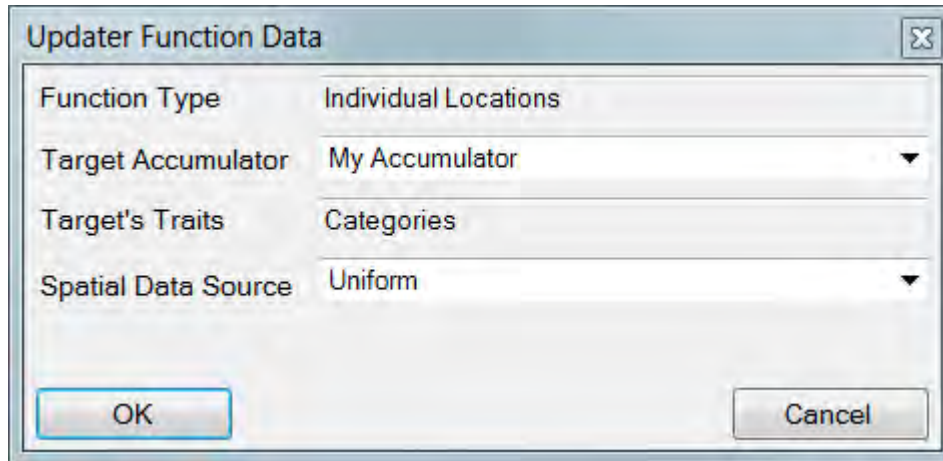
The Group Size updater function sets an accumulator equal to an individual's group size. Group members will always have a group size of at least one. Floaters will always have a group size of zero. The group size updater is used by the Floater/Group Status Trait Builder.

Increment



The Increment updater function adds to the selected accumulator. Decrementing an accumulator is accomplished by setting the updater's parameter to a negative value.

Individual Locations

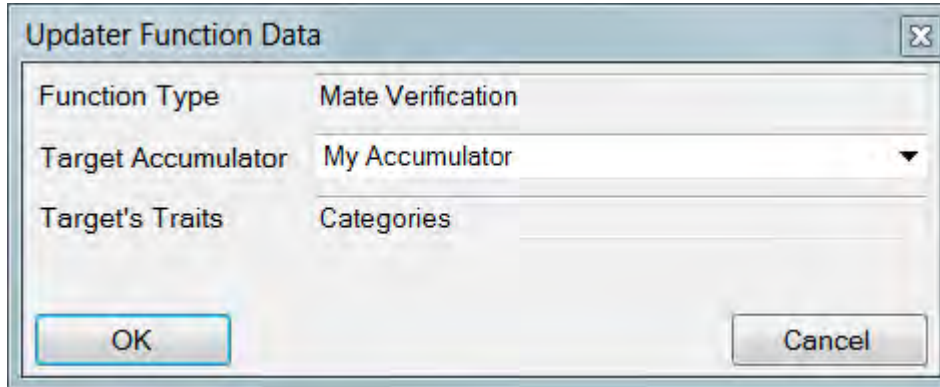


The Individual Locations updater function is used to identify an individual's location within the simulation landscape. Users select a target accumulator and a spatial data source. To use this updater, it is necessary to develop a HexMap in which each region of interest is composed entirely of hexagons with unique nonzero integer scores. For example, if there are 5 regions of interest, all of the hexagons in the first might be scored 1. Those in the second region of interest might be scored 2, and so on. Then an accumulated trait would be developed that assigned a "Region 1" trait value a threshold value of 1, and a "Region 2" trait value a threshold value of 2, etc. The very first trait value should be used to label individuals who are not in any region. The Individual Locations updater function would then be used to set the value of the accumulator linked to this accumulated trait. This way, when an individual is residing in a hexagon scored 1.0, it will end up being assigned the Region 1 trait value, and so on.

The updater begins by addressing each group. The score of every hexagon in a group's range is converted to an integer (ideally, the HexMap being used as the Spatial Data Source will have only integer values to begin with). Then the mode (most frequently occurring value) of these integer representations of the **non-zero** hexagon scores is assigned to the group member's target accumulator. Ties are settled randomly. Next, the updater function addresses each floater. Floaters already have a specific location, which is the last hexagon they dispersed to. That hexagon's score is converted to an integer, and assigned to the floater's target accumulator.

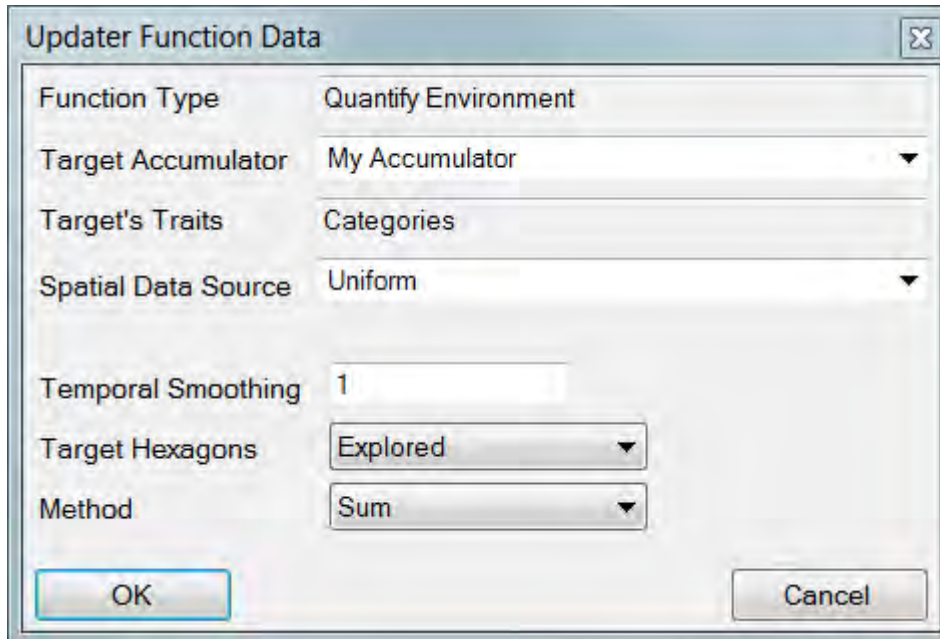
That is all the updater function has to do. The reason for converting hexagon scores to integers is that this allows the mode to be computed. The reason for computing a mode is that it allows group member's to be identified with a single location. If the Spatial Data Source contains real-valued hexagons, then the results may not be satisfactory.

When the mode of the range hexagons is computed, any zero-valued hexagons are left out of the calculation. This insures that individuals will be assigned to a location even if the majority of their range falls outside a specified location. The mode algorithm will only return a zero if every hexagon in the range is scored zero.

Mate Verification

The screenshot shows a dialog box titled "Updater Function Data" with a close button (X) in the top right corner. It contains three labeled fields: "Function Type" with the value "Mate Verification", "Target Accumulator" with a dropdown menu showing "My Accumulator", and "Target's Traits" with a dropdown menu showing "Categories". At the bottom, there are two buttons: "OK" and "Cancel".

The Mate Verification updater function is used to zero accumulators that hold the ID of a mate who has died. When reproduction depends on pair formation, it will be important to know when one member has died. This updater was designed specifically for this purpose. The updater does nothing if the mate is still alive. Otherwise it sets the mate ID to zero.

Quantify Environment

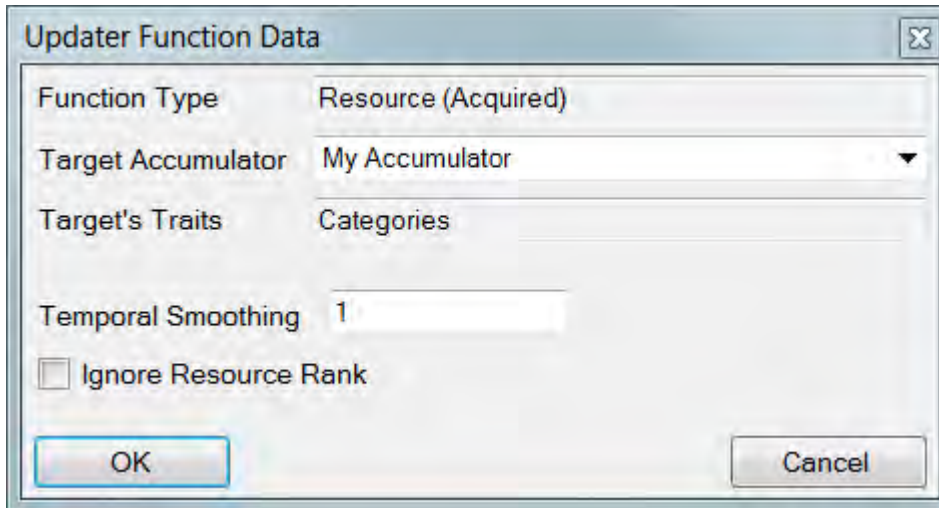
The screenshot shows a dialog box titled "Updater Function Data" with a close button (X) in the top right corner. It contains seven labeled fields: "Function Type" with the value "Quantify Environment", "Target Accumulator" with a dropdown menu showing "My Accumulator", "Target's Traits" with a dropdown menu showing "Categories", "Spatial Data Source" with a dropdown menu showing "Uniform", "Temporal Smoothing" with a text input field containing "1", "Target Hexagons" with a dropdown menu showing "Explored", and "Method" with a dropdown menu showing "Sum". At the bottom, there are two buttons: "OK" and "Cancel".

The Quantify Environment updater function is a flexible tool with many potential applications. Users select a spatial data time series to use in the quantification. The updater function then quantifies each individual's "exposure" to this spatial data. Exposure, in this sense, could be a good or a bad thing for the individuals in question. Three parameters control the computational details. Smoothing is covered earlier in this chapter. The second parameter allows users to choose whether to use individual's

allocated or explored areas to perform the computation. The third parameter tells the updater whether to report the mean value or the sum, taken across the hexagons in the individual's explored or allocated area.

Unlike the Acquired and Competitive Resource updater functions, Quantify Environment does not take resource availability into account. The environmental quality, or exposure, computed by this updater function does not change as a function of the number of individuals that are present at the site.

Resources (Acquired)



The screenshot shows a dialog box titled "Updater Function Data". It contains the following fields and controls:

- Function Type:** A dropdown menu with "Resource (Acquired)" selected.
- Target Accumulator:** A dropdown menu with "My Accumulator" selected.
- Target's Traits:** A text field containing "Categories".
- Temporal Smoothing:** A text field containing the number "1".
- Ignore Resource Rank:** An unchecked checkbox.
- Buttons:** "OK" and "Cancel" buttons at the bottom.

The Acquired Resource [labeled "Resources (Acquired)"] updater function returns the percent of an individual's target resource that it can derive from its range (group members) or allocated area (floaters). This computation is performed at the time the updater function is called. Thus if groups expand their ranges into hexagons previously allocated to floaters, the impact this has on the floater's ability to acquire resources from their allocation will be accounted for by the updater function. The four Acquired and Competitive Resource updater functions are the only ones that return a percentage, and users should keep this in mind when linking trait values to these accumulators. The reason for using the percent of an individual's target resource as the accumulator value is that individual resource needs can change in time. If an absolute resource quantity was tracked, and the updater was smoothing this value over two or more time steps, then the process of shifting from low to high resource needs would tend to drive individuals into a low resource acquisition class.

Because the intent is to measure resource use, the Acquired Resources updater function always uses the current Range Spatial Data to perform its computations. The Range Spatial Data is specified in the [Population Parameters Range Data Tab](#), but can be altered at any time using an [Adjust Range Parameters](#) event. This is also true for the Competitive Resources updater function, but it is not the case for the Quantify Environment updater function. The latter may be used to quantify spatial descriptions of

things other than resources, such as spatially distributed stressors. Thus the Quantify Environment updater function requires explicit specification of the spatial data to be used in its computations.

Resources are computed differently for floaters and group members. Group member resources are derived from ranges, and range resources can be shared equitably or based on individual resource ranks. Range resources may also be reduced by floaters if floaters are allowed to preempt group resources. see the Population Parameters Range Data Tab for further discussion of resource ranks and floater preemption. Floaters occupying range hexagons are allowed to divert a percentage of the hexagon's resources for their own usage (referred to as the "floater fraction"), but only a single floater can preempt any given range hexagon. Range resources become limiting when the sum of the group's resource targets exceeds the sum of the range hexagon scores (minus the floater fractions). [Range Dynamics](#) and [Floater Creation](#) events may be used to modify range resources through adjusting range structure (Range Dynamics) or adjusting group structure (Floater Creation). Groups may also expand their ranges when accepting immigrants during Movement events.

Floater allocations are comprised from a subset of the floater explored areas. Both explored and allocated areas are assigned by Movement events. Floater explored areas are established during both dispersal and exploration. Floater allocations are only assigned following exploration, when it's been determined that the individual has not been able to start or join a group. Floater explored areas may overlap, but any given hexagon's resources can only be allocated to a single floater at a time. A floater will claim all of the resources from each hexagon it has been allocated, unless that hexagon is part of a range. Floaters may only claim the percent of range hexagon's resources specified in the Floater Preemption field within the range parameters. Floater allocations are developed by selecting hexagons from a floater's explored area. These hexagons are added to the allocation if they have not already been claimed by another floater. This process continues until the floater's resource target has been reached, or until the entire explored area has been exhausted. The floater allocation is thus constrained by the distribution of ranges and other floaters in its vicinity at the time of its construction. The more floaters move, the more up-to-date their allocations will be.

The Acquired Resource updater has a Temporal Smoothing parameter, and an Ignore Resource Rank check-box. Smoothing is covered earlier in this chapter. The Ignore Resource Rank check-box allows users to ignore inequalities that may have been built into resource ranks. If the check-box is checked, then the resource ranks will be ignored and the total available resource will be divided up equally among all group members. When the check-box is unchecked, the resource ranks will be used, and resources will be handed out on a first-come first-served basis. The actual computations that are performed are as follows, with floaters being processed first:

Floaters

1. Randomize the list of floaters.

2. For each floater, add up the value of every hexagon in its allocated area. Floaters may only acquire a fraction (the preemption value) of hexagons owned by groups.
3. Divide the total value of the allocated area by the individual's resource target. Multiply this by 100% to get the new acquired resource value.

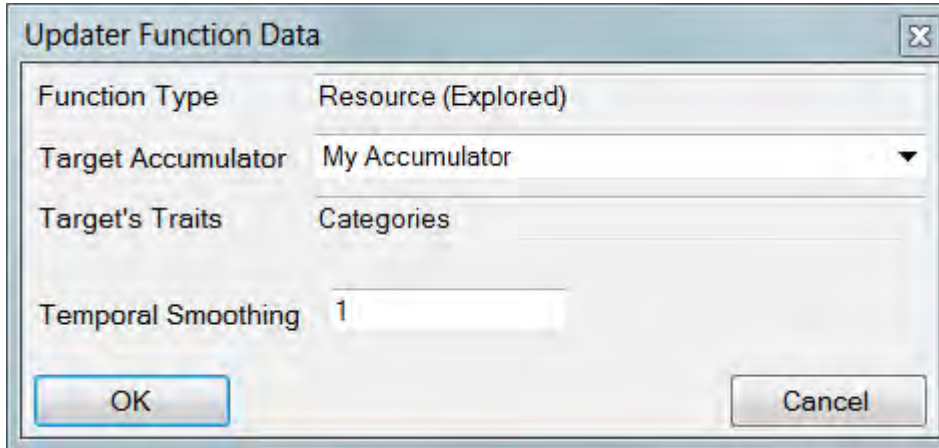
Group Members

1. Subtract the floater fractions (the preempted resources) from every range hexagon that is cross-allocated to a floater. This value is unavailable to the group members.
2. For each group, sum the adjusted (for floater fractions) value of all of the range hexagons. This is the total resource pool available to the group.
3. If the Ignore Resource Rank check-box is checked, then allocate the available resource pool equally among the group members. Then divide each individual's resource allocation by its resource target., and multiply the result by 100% to get the new acquired resource value. The algorithm is then finished.
4. If the Ignore Resource Rank check-box is not checked, then for each group, first randomize the list of group members, and second, sort the group members by resource rank. Then proceed one group at a time.
5. Select individuals one at a time, working from highest to lowest resource rank (within-rank individuals were randomized in step 1). Assign each individual its target resource (or the maximum possible if less than the target is available), and then remove this amount from the available resource pool. Continue this process until all group members have been attended to. Some group members may be allocated zero resources. This process does not guarantee that individuals within a resource rank will receive equal resource shares.
6. Divide each individual's resource allocation by its resource target. Multiply the by 100% to get the new acquired resource value.

The Acquired Resource updater function is the principal mechanism that user's have to stratify vital rates and behaviors based on resource availability. The main advantage of our implementation is that resource acquisition is distinguished from resource quality, the latter being a measure of the potential for resource acquisition that does not take conspecifics and competitors into account. Also, by smoothing the resource acquisition accumulators over time, users can insulate members of a population from sudden changes in vital rates that might otherwise result from a temporary excursion away from a resource-rich portion of the landscape. That is, individuals can effectively store resources in good times, and use these stores to carry them through lean times.

An Acquired Resource trait builder has been developed to simplify the process of modifying vital rates and behaviors based on resource acquisition.

Resources (Explored)



Updater Function Data

Function Type: Resource (Explored)

Target Accumulator: My Accumulator

Target's Traits: Categories

Temporal Smoothing: 1

OK Cancel

The Explored Resource [labeled "Resources (Explored)"] updater function is similar to the Allocated Resources updater function. When the Explored Resources updater is used, individuals acquire resources from their explored areas, not their allocated areas. When more than one individual has an explored hexagon in common, then the hexagon's resources are divided up equally among all individuals. The Explored Resources updater function does not differentiate between group members and floaters. Also, the Explored Resources updater function does not contain an Ignore Resource Rank check-box. Explored resources are always divided up equitably. For example, if 3 floaters, and 7 group members, from 2 different groups all share an explored hexagon, then that hexagon's resources will be divided into 10 equal shares, and distributed to the 10 individuals. Other than this difference, the Explored and Allocated Resources updater functions are the same.

For resource allocation to work with individual explored areas, there must be at least one Movement event in the event sequence that uses the Construct Explored Areas strategy. Movement strategies are selected from the Movement event's Global tab. The Construct Explored Areas strategy creates the Explored Counts data structure that is used to divide hexagon resources up among multiple individuals. If such a Movement event does not exist, then each individual will have access to all of the resources in its explored area, regardless of the extent to which explored areas overlap. This will unrealistically inflate of the amount of available resource.

If a single Movement event implements the Construct Explored Areas strategy, then that event must have the Clear Explored Counts check-box checked. If multiple Movement events use the Construct Explored Areas strategy, then the first one should have this box checked. The others should have the Clear Explored Counts box unchecked. The Clear Explored Counts check-box resets the Explored Counts data structure to zero. If this is not done once per time step, then counts will accumulate from time step to time

step, and will not reflect actual number of individuals competing for any one hexagon's resources.

Resources (Competitive Acquired)

The Competitive Resources updater function [labeled "Resources (Competitive Acquired)"] is a close analog of the Acquired Resources updater, which is described above. This Competitive Resources updater simply has the added ability to take competition between multiple populations into account. The resources available to a population experiencing competition may be less than the total that is actually present in the landscape. The Competitive Resources updater function can only work properly when two or more populations have been assigned a non-zero competitive ability. This is done using the Competitive Ability parameter (and check-box) in the [Population Parameters Range Data tab](#).

When the Competitive Resources updater runs, the first thing it does is to compute how much of the total resource is available to the target population. Once this is done, the updater functions exactly the same as the Acquired Resources updater. If there is no competition the two updater functions will produce identical results. If there are competitors present in the range, then the Competitive Resources updater may assign individuals a smaller percent resource than they would have obtained in the absence of competition.

The Competitive Resources updater function does not account for competition between floaters of different populations. For floaters, the Competitive and Acquired resource updater functions always produce the same results. Groups from multiple populations compete with each other, but floaters only compete within a population.

Competition is accounted for on a hexagon-by-hexagon basis. For every hexagon that is part of a range for k populations (where $k > 1$), the Competitive Resources updater partitions the hexagon's score into k separate "competition coefficients". This is accomplished as follows: Let C_i be the competitive ability of population i (this is specified in the Population Parameters Range Data tab), and let $N_{i,x}$ represent the number of individuals of population i in hexagon x . Population i will be assigned a competition coefficient for hexagon x equal to

$$\frac{C_i \cdot N_{i,x}}{\sum_k C_k \cdot N_{k,x}}$$

This coefficient is then multiplied by the hexagon score to obtain the hexagon's available resource for population i under competition. The competition coefficient for any given hexagon will be different for each population, but the sum for all competing populations is exactly 1.0 (even if the C_i terms do not add up to 100%). For floater

allocated areas, the competition coefficient is always one. The values for population size used to calculate the competition coefficients are gathered at the time that the updater function is run.

If two or more populations compete for resources, then those populations should all use the same Range Spatial Data. To do otherwise would produce meaningless results.

The Competitive Resource updater has a Temporal Smoothing parameter, and an Ignore Resource Rank check-box. Smoothing is covered earlier in this chapter. The Ignore Resource Rank check-box allows users to ignore inequalities that may have been built into resource ranks. If the check-box is checked, then the resource ranks will be ignored and the total available resource will be divided up equally among all group members. When the check-box is unchecked, the resource ranks will be used, and resources will be handed out on a first-come first-served basis.

Resources (Competitive Explored)

The Competitive Explored Resource [labeled "Resources (Competitive Explored)"] updater function is similar to the Competitive Acquired Resources updater function. When the Competitive Explored Resources updater is used, individuals acquire resources from their explored areas, not their allocated areas. When more than one individual from the same population has an explored hexagon in common, then the hexagon's resources are divided up equally among those individuals. The Competitive Explored Resources updater function does not differentiate between group members and floaters. Therefore Competitive Explored Resources can account for competition between both floaters and group members from multiple populations. Also, the Competitive Explored Resources updater function does not contain an Ignore Resource Rank check-box. Explored resources are always divided up equitably within a single population.

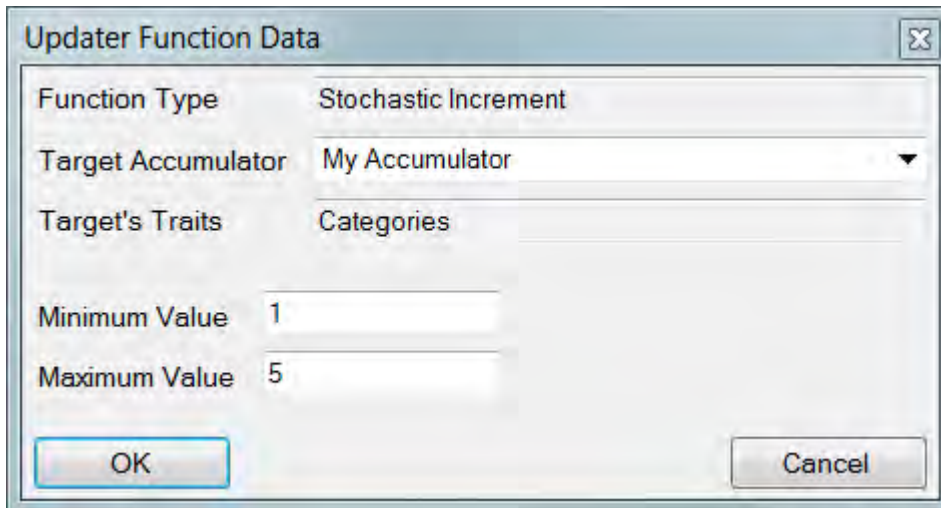
A significant difference between the Competitive Explored and Competitive Acquired Resources updater functions involves the way that competition between populations is handled. The Competitive Explored updater function does not take the magnitude of the Competitive Ability parameter into account. Instead, all individuals (group members and floaters) from all populations having a non-zero Competitive Ability parameter are treated as equal competitors. Thus the competition that does take place will be as if each population had an identical value for the Competitive Ability parameter. This is done to simplify the source code, but also because explored areas are not "owned" by the individuals that occupy them. Rather, they are just areas that have been recently visited. Explored areas may greatly exceed the space needed to sustain an individual.

Competition between populations can only be simulated if each competing population has been assigned a non-zero competitive ability. This is done using the Competitive Ability parameter (and check-box) in the [Population Parameters Range Data tab](#).

For resource allocation to work with individual explored areas, there must be at least one Movement event in the event sequence that uses the Construct Explored Areas strategy. Movement strategies are selected from the Movement event's Global tab. The Construct Explored Areas strategy creates the Explored Counts data structure that is used to divide hexagon resources up among multiple individuals. If such a Movement event does not exist, then each member of a population will have access to all of the resources in its explored area, regardless of the extent to which explored areas overlap with other individuals from the same population. This will unrealistically inflate the amount of available resource.

If a single Movement event implements the Construct Explored Areas strategy, then that event must have the Clear Explored Counts check-box checked. If multiple Movement events use the Construct Explored Areas strategy, then the first one should have this box checked. The others should have the Clear Explored Counts box unchecked. The Clear Explored Counts check-box resets the Explored Counts data structure to zero. If this is not done once per time step, then counts will accumulate from time step to time step, and will not reflect actual number of individuals competing for any one hexagon's resources.

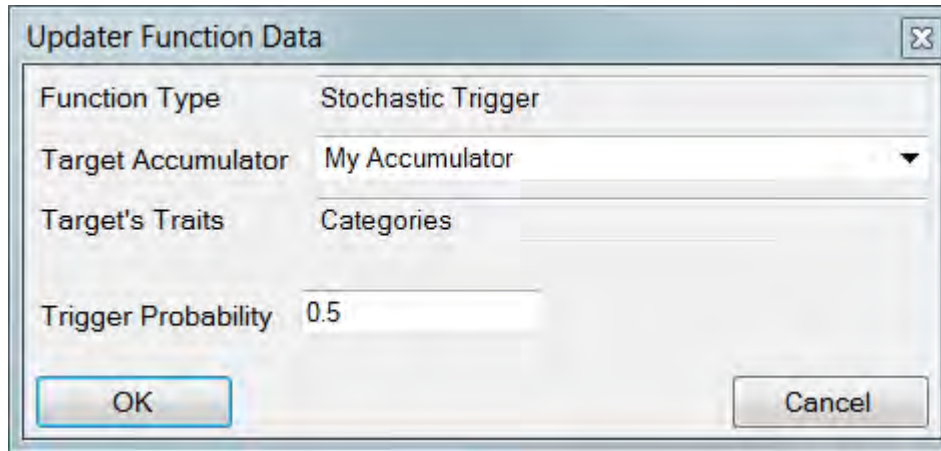
Stochastic Increment



Updater Function Data	
Function Type	Stochastic Increment
Target Accumulator	My Accumulator
Target's Traits	Categories
Minimum Value	1
Maximum Value	5
<input type="button" value="OK"/> <input type="button" value="Cancel"/>	

The Stochastic Increment updater function has two parameters, a minimum and a maximum value. When the updater runs, a uniformly distributed random variate is selected from the range given by [min, max], and added to the individual's accumulator. A separate random draw is made for every individual that is acted upon by the updater.

Stochastic Trigger

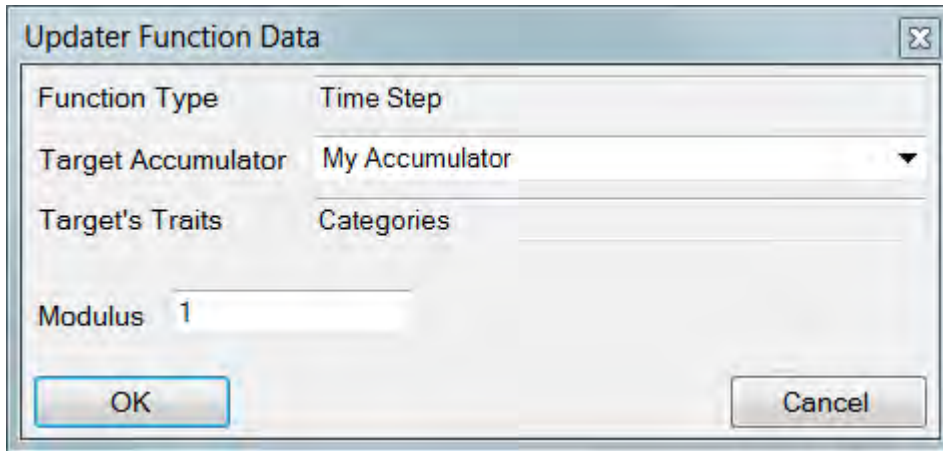


The Stochastic Trigger updater function returns a 1 with probability p (the value specified by the user) and a 0 with probability $1-p$. This updater function is unique in that it makes a single evaluation and then applies this result to every individual in the target population. Thus, every individual in the population will have this value in common. All of HexSim's other updater functions perform a separate evaluation for each individual.

The accumulator linked to the Stochastic Trigger updater function can be used to set a binary trait (e.g. "on" versus "off"). A HexSim life cycle event can then be stratified by this trait, and applied only to individuals for whom the trait value is "on". Effectively the result is that each time step the event will have a probability p of firing. So the real value of this updater function is the role it plays in making life cycle events that trigger probabilistically.

A Stochastic Switch trait builder exists exclusively to set up this machinery. Users are encouraged to take advantage of this trait builder as opposed to putting the parts together by hand. The Stochastic Switch Trait Builder creates an accumulator, and it adds an Accumulate event to the life cycle. The Accumulate event will be colored gray as a reminder that it was automatically generated. The accumulator uses a Stochastic Trigger updater function to set the switch off or on with the probability supplied to the Trait Builder. The accumulated trait switches from Off to On when the accumulator increases to one or more.

Time Step



The image shows a dialog box titled "Updater Function Data". It contains the following fields and controls:

- Function Type:** A text field containing "Time Step".
- Target Accumulator:** A dropdown menu showing "My Accumulator".
- Target's Traits:** A text field containing "Categories".
- Modulus:** A text field containing "1".
- Buttons:** "OK" and "Cancel" buttons at the bottom.

The Time Step updater function can be used to increment an accumulator on a time-step by time-step basis. This updater function has a modulus operator that can be used to invoke cyclic behavior. If the time step is represented by TS, and the modulus by M, then TS modulus M is equal to the remainder left after TS is divided by M. For example, as TS increases from 0 to 9, TS modulus 3 would equal: 0, 1, 2, 0, 1, 2, 0, 1, 2, 0. When the modulus is 1, the actual time step is always used.

The Adjust Range Parameters Event

The Adjust Range Parameters event makes it possible to alter the range parameters while a simulation is running. The Adjust Range Parameters event has exactly the same parameters and layout as the [Population Parameters Range Data tab](#). The only difference between the two is that the Adjust Range Parameters event also has a population selector. An image of the Adjust Range Parameters event window is shown below.

Competition | Adjust Range Parameters (Modify Group Size)

Name Modify Group Size

Properties Description

Population Competitor A

Range Spatial Data Matrix

Range Barriers >>> None Selected <<<

hectares & meters hexagons

Maximum Range Area 4250.0000 50.0000

Maximum Range Span 9907.0453 10.0000

Maximum Group Members 50

Hexagons Range-Eligible if Value is At Least 0.1000

Minimum Range Resource 50.0000

Floater Preemption of Group Resources (%) 0.0000

☒ Competitive Ability (%) 50.0000

Resource Targets

☐ Disturbance Class

☐ Fitness Class

☐ Resource Class

☒ Stage Class

Name	Rank	Target
Stage 0	0	10
Stage 1	0	20
Stage 2	0	30
Stage 3	0	40
Stage 4	0	40
Stage 5	0	30

Import

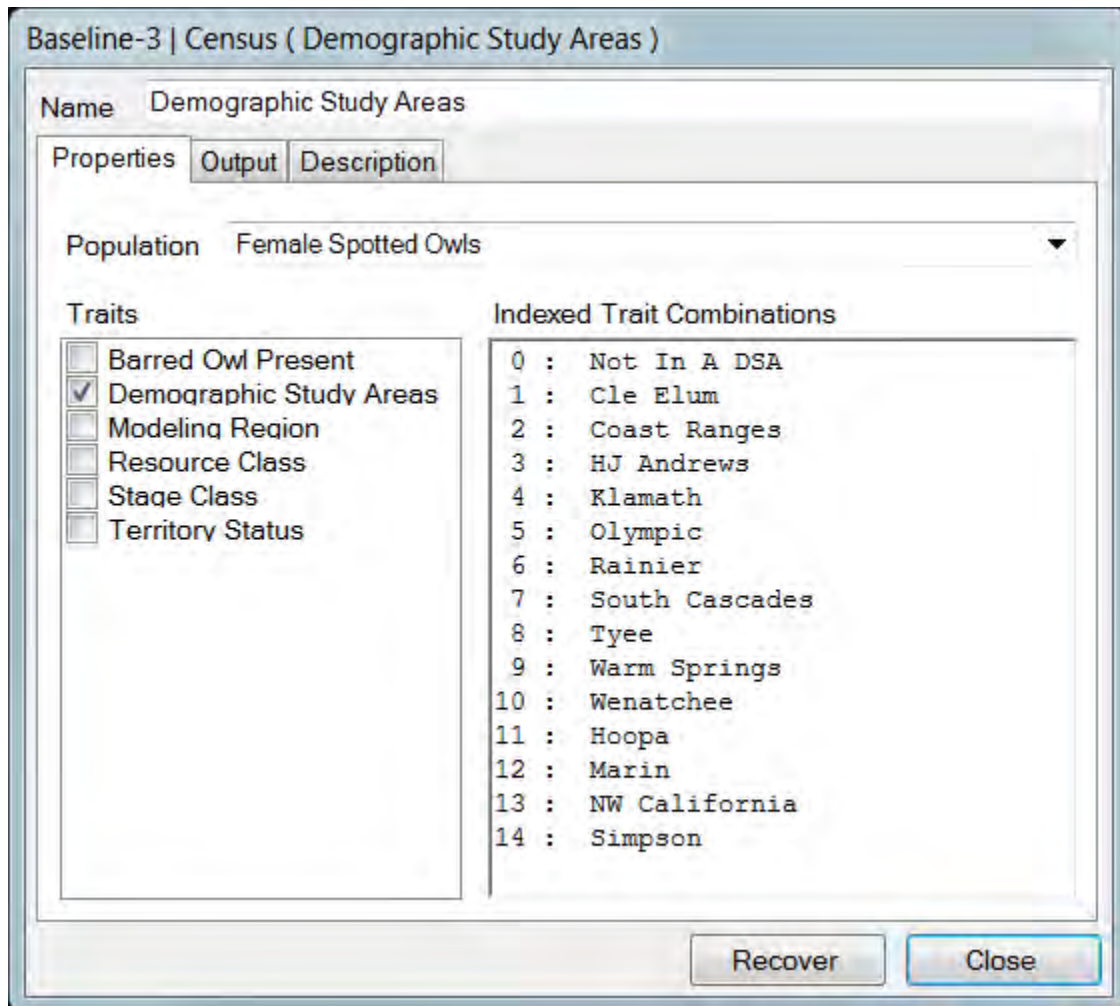
Recover Close

There are many potential reasons to alter the range parameters part way through the life cycle. Group structure or competition may change seasonally. Range sizes may

expand and contract, resource needs could increase or decrease, and so on. The range parameters specified in the Population Parameters Range Data Tab are referred to as the *static* range parameters. These static range parameters are restored at the start of every time step. When these are superceded by an Adjust Range Parameters event, the new values will remain in effect until they are either replaced by yet another Adjust Range Parameters event, or until the start of the next time step.

The Census Event

The Census event gathers population size data and writes it to a CSV file in the scenario's results folder. To parameterize Census events, users select a population and one or more traits. An image of the Census event is shown below.



Census event output files have six common data columns. These record the replicate number, time step, total population size, number of group members, number of floaters, and the instantaneous value of Lambda. The instantaneous lambda value for time step T is equal to $[\text{Population at } T] / [\text{Population at } T-1]$. This simple computation can be cumbersome to perform by hand, so its is generated as a convenience. The lambda value for time step zero is always set to one.

The remainder of the data written by the Census event is a list of all of the trait combinations that result from the trait or traits that have been selected. It is impractical

to supply meaningful headings for the potentially numerous trait value permutations. So HexSim simply titles each of these data columns with "Trait Index", followed by an integer value. The mapping between the trait indices and the trait value combinations is displayed in the Census event's Indexed Trait Combinations panel. As traits are selected or deselected, the index of trait combinations are updated.

It can be useful to insert multiple Census events into the life cycle. Each Census event will write its own output file. These CSV files will be distinguished by an integer suffix that appears just before the ".csv" in the file name. The first Census event in the life cycle will generate a file ending with ".0.csv". The second Census event will generate a file ending with ".1.csv", and so on.

The Census event data is gathered at the time the event fires. So users may, for example, want to place census events on either side of a Survival or Reproduction event. Doing so would quickly illustrate the impact the events were having on the population.

The Floater Creation Event

The Floater Creation event will take some or all group members, remove them from their groups, and make them floaters. The parameterization of the Floater Creation event determines which individuals stay and which are removed. The parameterization allows for floaters to be created based on the maximum allowed group size, trait values, and the available resources. An image of the Floater Creation event is shown below.

Baseline-3 | Floater Creation (Stage 0 Birds)

Name Stage 0 Birds

Global Output Description

Population Female Spotted Owls

Resource Threshold (%) 100.0000 All Groups Processed

☐ Enforce Maximum Group Size

Traits

- ☐ Barred Owl Present
- ☐ Demographic Study Areas
- ☐ Modeling Region
- ☐ Resource Class
- ☒ Stage Class
- ☐ Territory Status

Combinations

Name	Priority	% Target Resource
Stage 0	Always Float	100
Stage 1	Never Float	100
Stage 2	Never Float	100
Stage 3	Never Float	100

Recover Close

Floater Creation events use a priority ranking scheme to determine which individuals will stay, and which will leave a group. Priority ranks are always non-negative integer

values. A priority of 0 is used to indicate that an individual will always leave its group, thus becoming a floater. When a trait combination is assigned a priority of 0, the text "Always Float" is placed in the interface. The highest possible priority is equal to the largest unsigned integer the computer can represent. When this value is assigned, the text "Never Float" is placed into the interface. Always Float and Never Float are a special priorities because they force a specific result regardless of group size or resource availability. A context menu (see figure below) is provided that lets users directly select the Always Float and Never Float priority values. Integer priority values that fall in between these can be typed directly into the interface (see figure below).



When Floater Creation events run, groups are processed iteratively. However, a group will only be processed if its total range resources fall below a specified threshold value. This threshold value is set using the Resource Threshold parameter as follows. First, the group's cumulative resource demands are computed. This is the sum of each group member's resource target. Second, the cumulative range resources are computed. This is the sum of each range hexagon's score. Next, the cumulative range resources are divided by the cumulative resource demand, and the result is multiplied by 100%. Finally, this value is compared to the Resource Threshold parameter. If this value falls below the specified Resource Threshold, then the group is considered resource-deficient, and is processed. Otherwise it is skipped. The exception to this rule is that every group will automatically be processed if the Resource Threshold is set to 100%. The Resource Threshold parameter thus makes it possible to keep groups intact unless resource demands significantly outstrip resource availability.

Once it has been determined that a group will be processed, the members of the group are randomly ordered and then reordered according to their priority rank. Lower priority individuals are more likely to leave the group, and higher priority individuals are more likely to remain in the group. All other things being equal, a priority 1 individual is more likely to become a floater than a priority 2 individual. Users select one or more traits, and the priority values are stratified by the resulting trait value permutations.

The first action that the Floater Creation event takes is to remove all priority zero (Always Float) individuals from the group.

If the Enforce Maximum Group Size check-box is checked, then this condition is assessed next. If the group size exceeds the maximum allowed (set in the [Population Parameters Range Data Tab](#), or by an [Adjust Range Parameters](#) event), then individuals will be removed from the group until the group size decreases to the maximum allowed value. These removals are performed in order from lowest to highest priority, starting with priority 1 individuals (priority zero have already been removed at this point). However, individuals will not be removed from the group if the "Never Float" condition has been set for their trait combination. The Never Float priority will not be violated, even if it means that Floater Creation will be unable to reduce the group size to the maximum allowed value.

The final two steps of the Floater Creation algorithm make use of the "% Target Resource parameter". This parameter influences how much range resource must be assigned to individuals who remain in the group. Each individual's resource target is specified in the Population Parameters Range Data Tab, or by an Adjust Range Parameters event. The % Target Resources parameter specifies what percentage of this target resource an individual must be able to access to remain in the group. If an individual can access this much resource, then this resource will no longer be available to other group members (during this event). Setting the % Target Resource parameter to 100% means that affected individuals must acquire all of their target resources in order to remain in a group. Setting this parameter to zero means that affected individuals may stay in a group regardless of the available resources.

To begin, each range's total resource is tallied *without accounting for floater resource preemption*. This quantity is the sum of all of the range hexagon scores, as specified in the Range Spatial Data. Then this range-wide resource total is decremented by the resource use corresponding to each individual that has been assigned the "Never Float" priority. These individuals must remain in the group, and thus they must each reduce the total resource available to others by an amount equal to their $[\text{Target Resource}] * [\% \text{ Target Resource}] / 100$. The total quantity of range resource remaining after this step is complete could be zero or less.

Finally, resource availability is analyzed for the remaining members of the group. In descending order of priority (e.g. priority 5, then 4, then 3, etc), individuals attempt to claim resources from the range in an amount equal to their $[\text{Target Resource}] * [\% \text{ Target Resource}] / 100$. If they are successful, then they remain in the group and the remaining range resource available to others is reduced by this amount. If the requisite resources are not available, the individual is instead removed from the group. This process continues for all individuals not previously processed. Because resource targets may vary, some high priority individuals with large resource targets may end up leaving a group while lower priority individuals with lower resource targets end up staying.

Individuals who are assigned a % Target Resource value of zero will only be removed from the group if their priority is zero (Always Float), or if doing so is necessary to lower the group size to the maximum value allowed. A % Target Resource value of zero implies that the necessary available resource for the individual's remaining in the group is zero. And this condition will always be met (by convention, even when the available resource is zero or less).

Each floater generated by the Floater Creation event is assigned a random location within the range of the group it formerly belonged to. Individuals' explored areas are left unchanged, but their allocated area is set to zero.

To recap, the order of operations undertaken by the Floater Creation event is listed below. If the Resource Threshold is 100%, then these actions are imposed on all groups. Otherwise they are only imposed upon resource deficient groups.

1. Remove every individual with a priority of Always Float.
2. If Enforce Maximum Group Size is on, iteratively remove the lowest priority individuals until the group size is OK.
3. Compute the total resources available to the group.
4. Reduce the total available resource by the needs of the individuals with a priority of Never Float.
5. Working from highest to lowest priority, remove individuals whose resource needs can not be met. When an individual's resource needs can be met, reduce the total available resource by this amount.

The Generated Hexmap Event

Generated HexMaps make it possible for users to create novel spatial data from existing HexMaps and population occupancy data. All of the workspace spatial data are available for this purpose, plus occupancy data for each population being simulated. Each of these objects may be assigned a variable name, and then used to construct an equation. The equation describes the generated map. An image of the Generated HexMap event is displayed below.

Baseline-A | Generated Hexmap (Identify Nesting Sites)

Name Identify Nesting Sites

Properties Description

Hexmap Name Nesting Sites (cumulative)

Save every 10 time steps (zero = never save)

Add

Generator Input Names

h1	Nesting Sites (cumulative)
h2	Female Spotted Owls: Presence/Absence

Delete

Generated Map = h1 + h2

☐ Smooth Hexagon Values Method Mean

Ignore <= 0.0000 Radius 1.0000

☐ Use Scaling Min 0.0000 Max 0.0000

Recover Close

Users supply an event name, and a name for the generated HexMap. The available spatial and occupancy data are accessed through the pull-down menu to the right of the Add button. Use this menu to select a data source, and then hit the Add to assign the data to a variable. The variables are always labeled hx, where x represents an integer value. The occupancy data are gathered at the time the HexMap is generated. Occupancy is simply a binary record of presence and absence for the selected population. All hexagons in the occupancy map are initially set to zero. Then, for every

group, occupancy is set to 1 for each range hexagon. Occupancy by floaters is not tabulated.

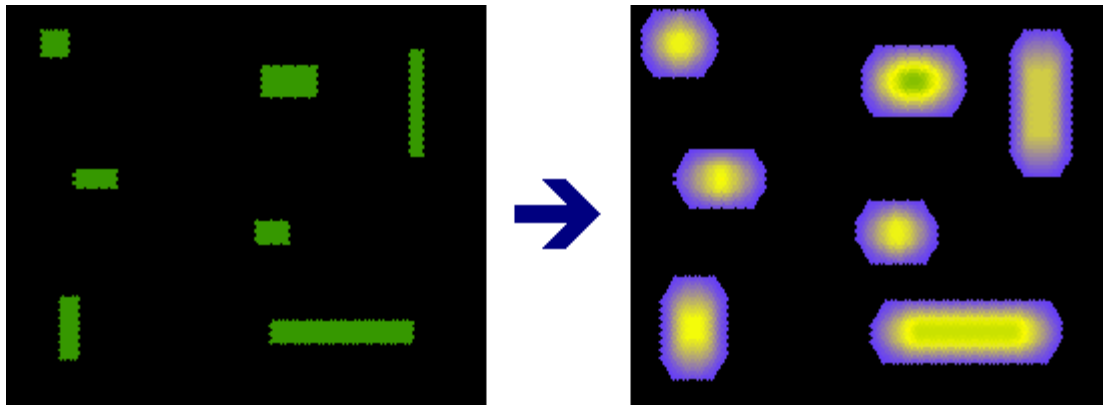
Once all of the desired variables have been created, you may construct a formula for the generated HexMap in the box at the bottom of the window. The formula will be applied in turn, for each hexagon in the grid. Maximum and minimum operators can also be added to these formulas. For example, the maximum value of variable h2 is represented by max2. The minimum value of variable h5 is represented by min5. These variables return the minimum or maximum values for the collection of hexagons.

The generated HexMap being created can itself be used as a variable in the map generation process. This might be used to decrement resources based on occupancy rates, or to keep a record of commonly used breeding sites, etc. If such recursion is implemented, a separate Generated HexMap event can be used to initialize the generated HexMap. This initialization Generated HexMap event will typically be set to trigger only on time step 1. Uninitialized generated HexMaps will be set to all zeros when they are created.

HexSim allows generated HexMaps to be written to the results folder. They can subsequently be viewed, which provides useful feedback into the workings of the Generated HexMap event. But writing generated HexMaps will take time, and will consume disk space. So the Generated HexMap event allows users to specify an interval for writing the data to the results folder. Once created, the generated HexMaps can be viewed using the Display Spatial Data tool, which is accessed through the HexSim drop-down menu.

The Generated HexMap event also allows users to smooth and rescale the maps it creates. This will be particularly useful when the generated HexMap is being used to guide dispersal. Smoothing a HexMap makes it possible for dispersing individuals to walk up-gradient towards desired targets. Otherwise, the dispersers must stumble upon a target to succeed. Up-gradient movement can be achieved using the Repulsion / Attraction feature of the dispersal component of Movement events. But attraction and repulsion are parameterized using absolute hexagon scores, and this is where the scaling algorithm comes into play. Its hard to know in advance what the range of values will be for a generated HexMap. But the range of values for a rescaled HexMap will be known. Thus the attraction and repulsion parameters can be set appropriately.

The smoothing algorithm can also be accessed from the spatial data context menu in HexSim's Workspace tab. An illustration of HexMap smoothing is provided below.



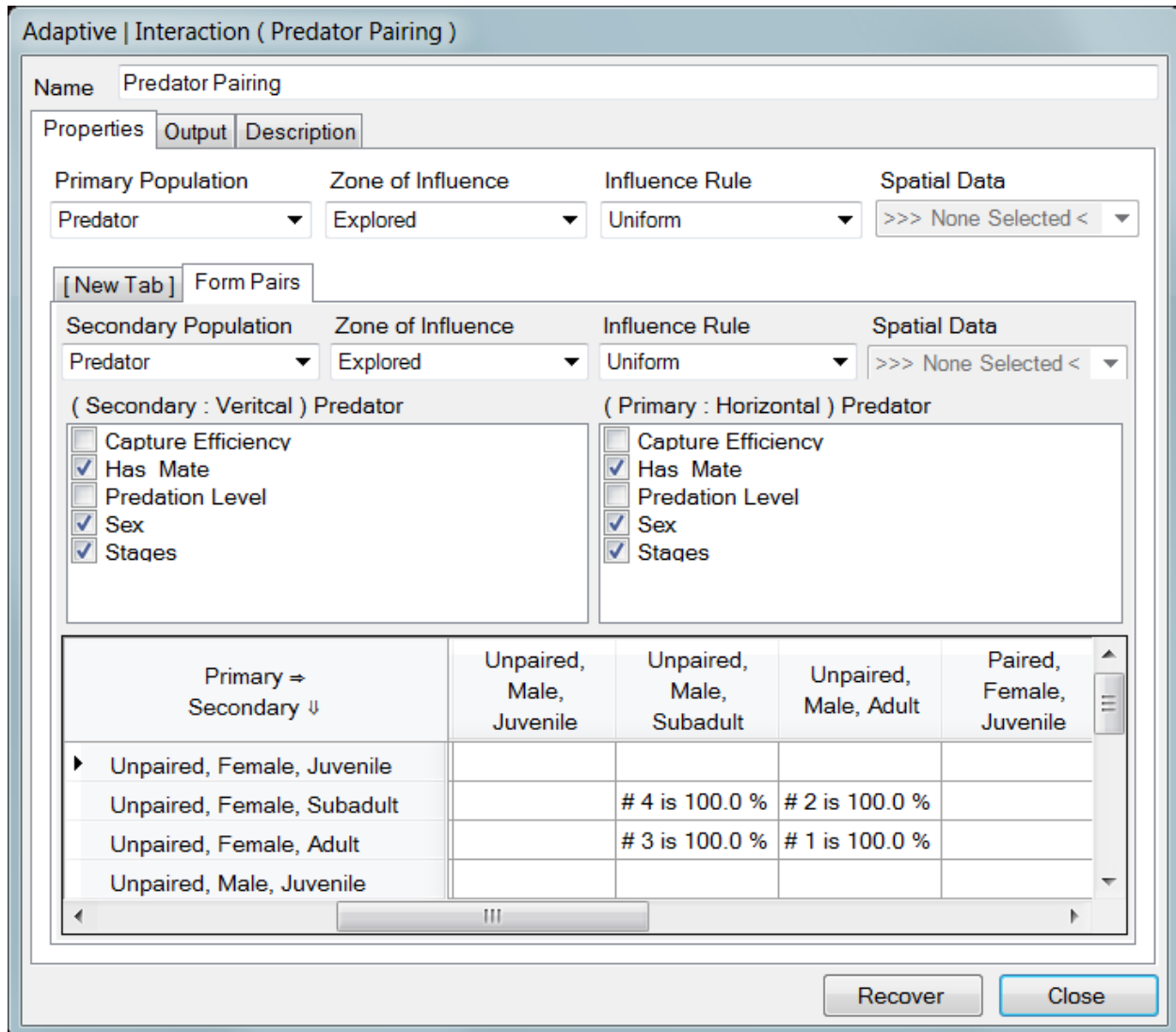
The smoothing algorithm allows users to select either a Mean, Mode, or Sum operator. A radius is also selected. HexSim uses a moving window approach to smoothing, whereby each hexagon is assigned a value equal to the mean, mode, or sum of all of the hexagons falling within the window. The window is always centered on the hexagon being evaluated, and the window radius is set to the value specified in the interface. The moving window will be truncated when HexMap edges are encountered.

HexSim's smoothing algorithm has one additional parameter that allows users to skip hexagons that fall below a threshold value. This is the parameter labeled "Ignore \leq ". If a hexagon is ignored, that means both that it will not be smoothed, and it will also never be used to compute a smoothed value for another hexagon.

It is possible to set up a life cycle so that an event X that depending on a generated HexMap precedes the Generate HexMap event. On the very first time step, this event X will attempt to access the Generated HexMap, but it will not yet have been created. This will cause HexSim to crash.

The Interaction Event

The Interaction event is probably the most complex of all HexSim events. But it is flexible, and not all that difficult to master. Interactions are always pair-wise, but may take place between two separate populations, or between members of the same population. Interaction events can be used to simulate predation, pair formation for purposes of mating, disease transmission, to force group-joining, and for a number of other issues. Individuals may interact if they co-occur in space, and the user has control over how this co-occurrence is evaluated. Since not all individuals who meet will interact in the same way, or interact at all, both the primary and secondary populations are trait-stratified. The permutations of the selected traits are used to assemble an interaction outcomes matrix. This structure allows users to define the nature of the interactions. These outcomes can vary as a function of the states of the individual actors. An image of the Interaction event is shown below.



The Properties tab has two rows of parameters at the top of its interface, one for the primary, and one for all secondary populations. The New Tab button is used to create one or more secondary population tabs. At least one secondary population tab must be added before an Interaction event can be fully parameterized. The primary population interacts with each secondary population. Each secondary population is represented by a unique data tab in the interface. The order of the secondary population tabs (from first added, to last added) defines the order in which they will be operated upon by the primary population. That is, if there is more than one secondary population, then the primary population will initially interact with the first (left-most) secondary population. Next, the primary population will interact with the second secondary population, and so on. The last secondary population to be interacted with will be the one in the right-most tab.

Any two individuals can only interact if they meet. HexSim determines whether individuals meet by comparing their "zones of influence". Each population has a Zone of Influence and an Influence Rule. They also have a Spatial Data menu, but it is only

active when the Influence Rule is set to "Weighted". The Zone of Influence can be set to an individual's explored or allocated area. The Influence Rule can be set to Always, Uniform, or Weighted. *Always* means that individuals from the corresponding population can be thought of as always present *throughout* their zone of influence. *Uniform* means the probability that an individual is present in a subset of its zone of influence equals subset's relative size. That is, if you consider a portion of the zone of influence that was 10% of the whole, then under the Uniform rule, an individual would have a 10% chance of being found there. *Weighted* means that the likelihood of being in a region within the zone of influence is proportional to its area, but weighted by quality. In this case, quality is derived from the spatial data selected by the user. This option simulates a preferential use of high quality areas within an individual's zone of influence.

The probability that a member of the primary population meets a member of the secondary population is derived from the two zones of influence, and their associated rules. A couple of examples will help make this clear. Assume that X and Y are members of the primary and secondary population. Lets say that X's influence zone is made up of 10 hexagons, and Y's is made up of 30. Now assume that the two zones of influence have 6 hexagons in common. This means that 60% of X's zone of influence overlaps with Y's, and 20% of Y's zone of influence overlaps with X's. Now lets say that both populations were assigned a Uniform influence rule. The probability that the two individuals would meet is 0.60 (the probability that X is in the overlapping region) multiplied by 0.20 (the probability that Y is in the overlapping region). Therefore the probability that they will meet is 0.12.

Now, if we take the above example, but set Y's Influence Rule to Always, then the probability that Y will be in the overlapping region becomes 1.0. Thus the probability of interaction becomes 0.6. When the Influence Rule is set to weighted, the probability that the individual is going to be in the area of overlap is equal to:

$$\sum_{i=1}^k Q_i \div \sum_{i=1}^n Q_i$$

where $i = [1, k]$ are the hexagons in the area of overlap, $i = [k + 1, n]$ are the hexagons outside of the area of overlap, and the Q_i is the quality of the i^{th} hexagon.

The middle portion of the Interaction event's Properties tab is used for trait selection. Trait selectors are supplied for both the primary population and for the secondary population that was assigned to this interactions tab. More than one trait can be selected simultaneously. The trait selections are used to assemble an "Outcomes Matrix". The outcome matrix's columns list the primary population's selected trait combinations, and the columns list those of the secondary population. If no traits are selected for a population, then only a single row or column will be generated in the outcomes matrix, and this row or column will apply to all members of the respective population.

The outcomes matrix allows users to specify what consequence ensues when members of the primary and secondary populations meet. For some trait combinations there may be no interaction at all. For example, if predators and prey both vary in size, then the interaction matrix might be set up to ensure that large predators would consume small prey, but not the other way around. And different outcomes could, for example, differentiate between the energy gain attributed to the consumption of small versus large prey items.

Interaction outcomes can be specified by right-clicking in one of the interaction outcomes matrix cells, or by double-clicking. Doing so brings up the Interaction Outcomes parameterization window, which has separate panels for the primary and secondary populations. When an outcome has been created, that cell in the interactions matrix will be labeled with an order number and an interaction probability (see figure above). Interactions do not need to be defined for every cell in the interactions matrix. Empty cells simply correspond to trait combinations that do not result in interaction outcomes.

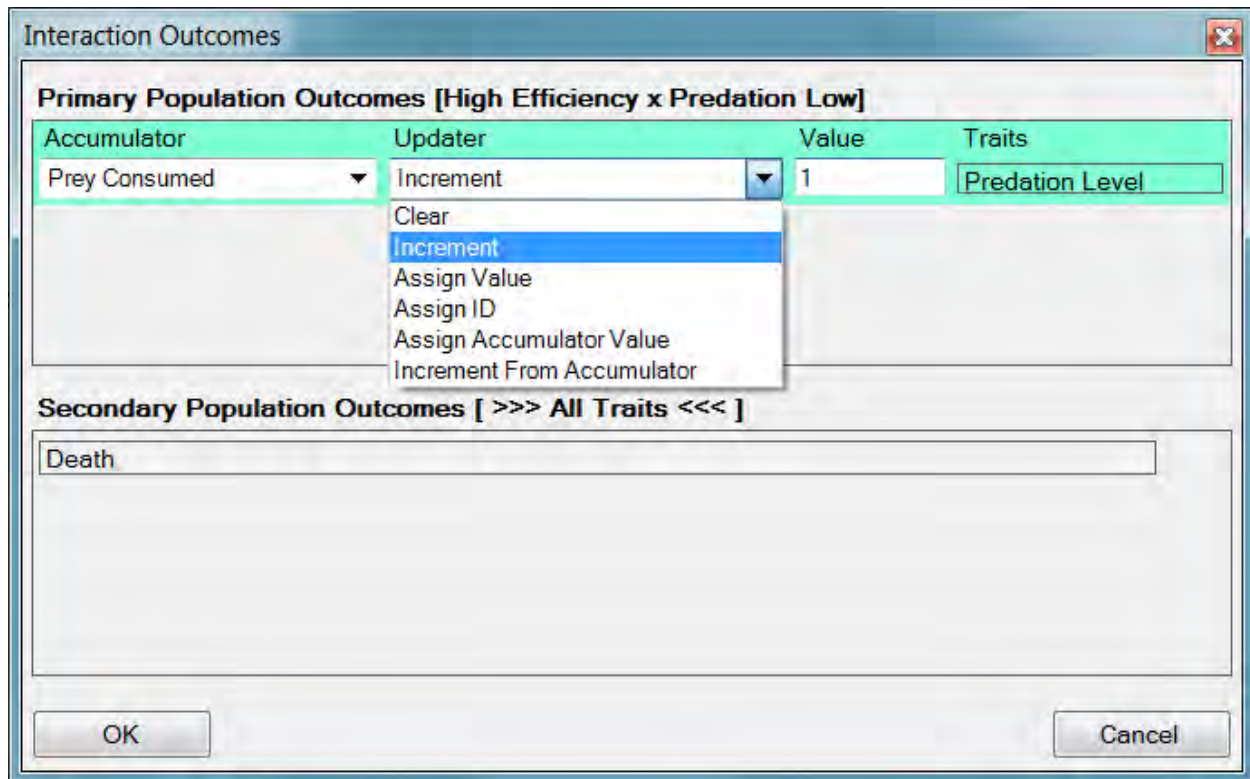
When interaction outcomes are added to the outcomes matrix, they are assigned a sequential order number (see figure above). The order number indicates what order the outcomes will be evaluated in. If outcomes may be mutually-exclusive, and some are more likely than others, then the more probably outcomes should be assigned lower order numbers. There is no way to change the outcome order numbers. But there is a cut and paste facility (see the context menu) associated with the outcomes matrix that can help users reorder them.

Interaction outcomes also come with an interaction probability (see figure above). The interaction probability can be changed using the outcomes matrix's context menu. The interaction probability is the likelihood that there will be an outcome when any two individuals do interact. If there is no outcome, then nothing happens to either individual. Outcome probabilities are evaluated separately (that is, a new random number is drawn and compared to the outcome probability) for each pair of interacting individuals.

Thus there are three different criteria that have to be met before an interaction outcome is actually experienced. First, two individuals must meet. This is determined by their zones of influence and influence rules. Second, the individuals must have the ability to interact. Interactions will only be possible the two individuals they have trait values that correspond to an outcome in the outcomes matrix. Third, an outcome must result from the interaction. This is evaluated based on the outcome probability.

The Interaction Outcomes dialog has a panel (the upper one) for the primary population and another (the lower one) for the secondary population. Outcomes are added to these panels using their context menus. The context menu has options for adding and deleting outcomes, and also for setting two special outcomes: death, and group-joining. Outcomes do not have to be added for both the primary and secondary population. Multiple outcomes can be associated with either population. But if death is selected as the outcome, no other outcomes can be added. If an outcome is placed into the Primary

Population Outcomes panel, then this outcome will affect members of the Primary Population, and likewise for the Secondary Population. An image of the Interaction Outcomes dialog window is shown below.



The death and group-joining outcomes have no parameters to set. The consequences of death are intuitive. Group-joining is a way to force a specific individual to join another's group, regardless of resource constraints. It is anticipated that this will be used principally in conjunction with the Assign ID updater function to simultaneously identify a mate and bring it into the group. When one individual joins another's group, it actually relocates to that group. Its explored and allocated areas get set to the new group's range. If it was the last remaining member of its old group, then the old group and range are deleted.

Custom outcomes necessitate specification of an accumulator and an accumulator updater function. Some updater functions require an additional parameter. The available accumulators are simply those supplied to the population in the [Population Parameters Traits Tab](#). Once an accumulator is specified, the corresponding traits are displayed in a read-only column at the right of the outcomes panel. The accumulator updater functions include Clear, Increment, Assign Value, Assign ID, Assign Accumulator Value, and Increment From Accumulator.

The Clear, Increment, and Assign Value updater functions all operate directly on the target individual's updater. The Assign ID, Assign Accumulator Value, and Increment From Accumulator updater functions do so as well, but the value that is assigned is

taken from an accumulator belonging to a member of the other population. The updater functions have the following behaviors:

- The Clear updater simply sets the selected accumulator to zero.
- The Increment updater adds to the selected accumulator. The amount to be added is specified in the field labeled "Value". Accumulators will be decremented when they are incremented by a negative quantity.
- The Assign Value updater assigns a value to the selected accumulator. The value to be assigned is specified in the field labeled "Value".
- The Assign ID updater assigns the other individual's ID to the selected accumulator. That is, one member of the interacting pair gets the other member's ID, which is assigned to the selected accumulator.
- The Assign Accumulator Value updater takes the value from a source accumulator (belonging to the other member of the interacting pair), and assigns it to the selected target accumulator.
- The Increment From Accumulator updater takes the value from a source accumulator (belonging to the other member of the interacting pair), and adds it to the selected target accumulator.

The Primary and Secondary population outcomes might be very different. For example, if the primary population is a predator and the secondary population is prey, then the predator might update an accumulator that tracks energetics, and the prey might simply die. Alternatively, an outcomes matrix might be developed to track disease transmission. Rows and columns could be labeled Susceptible, Exposed, Infected, Recovered. Then interaction outcomes could be specified to cover just the cases in which susceptible and infected individuals meet. All other meetings might be inconsequential.

When an interaction outcome changes the trait values for one or both member of the interacting pair, those changes are implemented immediately. For example, assume an Interaction event is being used to pair females with males. Suppose a "Has-Mate" trait with values "No" and "Yes" is used to track individual's pairing state. If the outcomes matrix is set up to only act on individuals with a Has-Mate trait value of No, then only a single pair will form -- even if an unpaired female meets multiple unpaired males in the same Interaction event. This is because as soon as the first pair is formed, the female's Has-Mate trait will switch to Yes. Her subsequent encounters with a other unpaired males will correspond to a paired-unpaired cell in the outcomes matrix, which will be empty.

The Introduction Event

The Introduction event adds new members to a population. Users have control over the number of individuals that are added, their placement, and their trait values. The Initial Placement is performed exactly as it is during simulation initialization, which is described in the section on the [Population Parameters Properties Tab](#). But the Introduction event allows users to select spatial data for initialization that are different from that used to start the simulation.

The introduced population must be assigned Trait and Accumulator values. Again, this is done exactly as it is during simulation initialization, using the data specified in the Population Parameters Traits Tab. However, the Introduction event allows users to override the trait values for probabilistic traits. A check-box is listed for each probabilistic trait. When one of these boxes is checked, the corresponding drop-down menu becomes active. This menu allows users to select a trait value that will then be assigned to all members of the introduced population.

An image of the Introduction event is displayed below.

Basic 3 | Introduction (Add Flagged Individuals)

Name Add Flagged Individuals

Properties Output Description

Population Test Population

Individuals 100

Placement Initialization

Probabilistic Trait	Force to Value
<input checked="" type="checkbox"/> Introduced	YES

Recover Close

Users are not allowed to override the values of accumulated or genetic traits because this would create a mismatch between the trait value and accumulator, or genotype. Probabilistic traits, in contrast, are independent of any other data stored within an individual.

If users want to introduce individuals with a specific accumulated trait, then the Introduction event should set a binary probabilistic trait that flags those individuals. For example, every individual at model initialization, and at birth, could be assigned a probabilistic trait named "Introduced" with a value set to "NO". The introduced population however, could have this trait forced to "YES". If no other events ever altered this trait value, then the introduced population would always be easy to identify. A Transition event could then be added subsequent to the Introduction event. This Transition event could modify an accumulator, but be stratified so that it only acted on

individuals who had the Introduced trait set to YES. The result will be an introduced population that has a custom accumulated trait value.

If users want to introduce individuals with a specific genetic trait, then this can be done by taking advantage of the Use Regional Distribution feature within the Locus Data dialog window. See the section on [HexSim Genetics](#) for more information on the use of regional allele distributions. This can be accomplished as follows. When the initial population is placed in the model at startup, they should be placed into a specific portion of the landscape. When new members of the population are introduced later on, they should be placed in a separate non-overlapping portion of the landscape. Special HexMaps can be developed and used to locate the initial and introduced populations correctly, and also to set their genotypes via the regional distribution tool.

For example, suppose the initial population is all to have genotype X, while the introduced population is to have genotype Y. The initial population would be placed in an area identified as non-zero in a binary HexMap called HexMap-X. This would be accomplished by setting the Initial Population Placement field to HexMap-X in the Population Parameters Properties tab. The introduced population would be placed in a distinct area identified as non-zero in a binary HexMap called HexMap-Y. This would be accomplished by setting the Placement field to HexMap-Y in the Introduction event. Then a third HexMap, HexMap-XY would be created that merged HexMap-X and HexMap-Y. The nonzero HexMap-X hexagons would be assigned one score (e.g. 1), and the nonzero HexMap-Y hexagons would be assigned a separate score (e.g. 2). The Use Regional Distribution check-box, in the Locus Data dialog window, would be checked, and HexMap-XY would be used as the Spatial Data Series. The Number of Intervals would be set to 2, and the initial population allele frequencies would be assigned to the first interval, and the introduced population allele frequencies assigned to the second interval.

Movement Basics

HexSim's Movement event is designed to be flexible. There are three principal actions that a Movement event can perform. First, it can be used to relocate individuals. Second, it can construct explored and allocated areas. Third, it can build groups and ranges. These three actions are closely related, but it is important to understand their individual significance. The first is obvious; Movement events move individuals. However, with one exception that is addressed below, Movement events only act on floaters. If you want a group member to move, it must first abandon its group. This is accomplished using a Floater Creation event. Every individual has an explored and an allocated area, although these can be empty at times. An individual's explored area is a record of the set of hexagons it has most recently dispersed through, or searched in hopes of starting or joining a group. The goal of movement is typically for individuals to build new groups and ranges, or to join existing groups. There is an option in the Interaction event that forces one individual to join the other's group. With that exception, group joining is conducted exclusively by Movement events. The Movement event provides the only mechanism for initiating new groups.

Movement events require specification of a population and a set of traits and trait combinations. Together these parameters determine who moves and who does not. Movement is broken down into Dispersal and Exploration. Dispersal in HexSim is the process of moving across a landscape by taking a series of steps from one hexagon to another. Users have control over how many steps are taken, and the choice of hexagons to move into can be biased by hexagon quality and an auto-correlation parameter. Dispersers can be made to stop early if they pass through particularly good portions of a landscape. Also, dispersal can be used to move individuals towards a hexagon that was visited in the past and remembered. Dispersing individuals are cognizant of landscape quality, but not of resource availability. Thus more dispersing individuals may be attracted to good locations in a landscape than that area might be able to support.

Exploration in HexSim involves carefully examining a portion of a landscape in hopes of starting or joining a group. Individuals who are exploring an area will be cognizant of both hexagon quality and resource availability. There are multiple primary and secondary goals that can be used with exploration. Explorers will search around themselves in hopes of attaining their goal. If the goal cannot be attained, then the individual will remain a floater. It is, however, simple to ensure that all individuals are floaters or all are group members, and it is straightforward to move individuals back and forth between these states.

Movement spatial data can be defined differently for dispersal and exploration. Dispersal spatial data is supplied directly within the Dispersal tab. The range spatial data is always used for exploration. This is done because exploration is the process

whereby ranges are formed. Range spatial data is defined in the [Population Parameters Range Data Tab](#), or by an [Adjust Range Parameters](#) event.

Movement barrier spatial data is set in the Movement event's Global tab. Barriers defined here will affect both dispersal and exploration in the same way. Range barriers are defined separately in the Population Parameters Range Data Tab, or by an Adjust Range Parameters event. Regardless of their parameterization (i.e. mortality or deflection probabilities), Range Barriers have only a single consequence. The exploration process will not be allowed to construct a range that is entirely bisected by a Range Barrier. Unlike movement barriers, range barriers have no impact on dispersal paths or exploration patterns.

Mortality spatial data may also be supplied to Movement events. This is done in the Movement event's Global tab. When used, mortality spatial data will affect both dispersal and exploration. Mortality spatial data is used to establish a per-step probability of mortality. Each time a hexagon is entered, be it from dispersal or exploration, that hexagon's score in the Mortality spatial data will be read and used as a probability. If a uniform random number drawn from $U[0, 1]$ is less than this value, then the individual will die. Needless to say, only carefully constructed spatial data should be used for this purpose. As a precaution, a Mortality check-box forces users to take an additional step before they specify Mortality spatial data.

Movement in HexSim involves setting strategies and goals. Strategies control the overall behavior of the Movement event itself. Available strategies include Translocate Then Explore, Dispersal Then Exploration, Exploration Then Dispersal, Dispersal Only, Exploration Only, and Construct Explored Areas. In addition, users may specify a Maximum Number of Explorations parameter. If a movement strategy involves both dispersal and exploration, then it must always end with an exploration. This ensures that those individuals who move get a chance to reach their goal.

Translocation can be used to simulate migration or the displacement of organisms by some external force (e.g. capture and relocation). When this strategy is used, a Translocation tab is created. This new tab is used to specify the Translocation spatial data. Each individual that is translocated will select one non-zero hexagon at random from the Translocation spatial data, and move to that site. After translocation, each individual takes one or more explorations. No mortality will result from translocation. If users want to focus the translocation destinations into one or more well-defined areas, the Translocation spatial data can be made mostly zeros, with just a few non-zero hexagon targets. The exploration that follows translocation will use the range spatial data (as is always the case for exploration).

The Construct Explored Area strategy is unique in that it operates on both floaters and group members. This is the only movement strategy that targets group members. This strategy is designed to allow individuals to explore without dispersing first, and in the case of group members, without giving up their group and range. For floaters this is no different than the Exploration Only strategy -- both will update a floaters explored and

allocated areas. But for group members, this provides the only way to modify their explored area without first becoming a floater (which would entail giving up both their group and range). This strategy is useful as a precursor to an Interaction event. For example, a predator might have constructed a small range in a good location within its range spatial data. It may then want to search broadly around this range for prey. The Construct Explored Areas strategy will allow the predator to generate a large new explored area without abandoning its existing range. And this explored area might be constructed using a generated HexMap that combines landscape features with prey occupancy data (this would entail using a [Generated HexMap](#) event followed by an Adjust Range Parameters event to assign the generated map to the range spatial data). Next, an [Interaction](#) event could be used to "capture" prey that fell within the new explored area.

The remaining strategies involve combinations of dispersal and exploration. Multiple explorations may be conducted. If this done, the event will always finish with an exploration to ensure that individuals get a chance to attain their goal. (Goals cannot be attained when movement ends with dispersal.) When multiple dispersal and exploration pairs are used, the dispersal path will avoid moving into hexagons visited during the most recent exploration. This feature is included under the assumption that this sort of redundancy would be disadvantageous in nature. However, HexSim does not attempt to store the data from earlier explorations. So an individual might disperse through hexagons that were explored two or three cycles back. The use of multiple dispersal and exploration pairs allows users to simulate a process whereby individuals make a limited attempt to acquire resources, and upon failure simply move to another part of the landscape and repeat -- all within the same event. The alternative would be to explore a much larger area, which is computationally inefficient and in many cases biologically unrealistic. In the figure below, the black lines depict dispersal, and the yellow hexagons depict explored areas.

Exploration goals are defined and executed in the exploration component of Movement events. HexSim's exploration goals include starting new groups, joining existing groups, and permutations such as "Start New else Join Existing". When resources are limited, the success of one individual could limit the prospects for another. For this reason, users may specify that individuals move in the order in which they were ranked in the Population Parameters Range Data Tab, or by an Adjust Range Parameters event. Higher ranked individuals will get to move first, and therefore will be more likely to obtain limited resources.

When the primary goal is to join a group, each newly explored hexagon is evaluated to see if it is part of a group's range. If so, then a determination is made regarding whether this group may be joined. A group may be joined if doing so will not exceed the maximum group size, and if its range has adequate resources to meet the specified percentage of the group's collective resource targets, including those of the immigrant. The group being joined is allowed to expand its range in order to accommodate the newcomer. If the group can be joined, then the individual's move will end. Otherwise the individual will continue exploring.

When the primary goal is to start a group, each newly explored hexagon is added to one or more cached ranges, which are then examined to see if any attain the individual's resource target. If one or more do, then a new group is formed and assigned the best cached range. Otherwise, the exploration process continues. If the individual reaches the maximum allowed exploration area without identifying a range that meets its resource target, then it is allowed to start a group with a resource-deficient range. Still, all ranges must adhere to the constraints present in the Population Parameters Range Data Tab, or by an Adjust Range Parameters event.

Two of the exploration goals are slightly more complex. Start New else Join Existing New and Join Existing New else Start New both make use of the concept referred to as an "Existing New" group. In HexSim, an Existing New group is one that was created during the current movement event. This distinction is useful when it is desirable to form groups comprised only of those individuals who actively moving. Without these goals, it would not be possible to restrict group joining to newly formed groups.

If an individual is unable to achieve its primary goal, and joining a group is the secondary goal, then the explored area is examined to see if it contains groups that can be joined. The hexagons are evaluated in reverse chronological order (the most recently explored hexagon is evaluated first). Each hexagon in the explored area is examined to see if it is part of a group's range. If so, a determination is made regarding whether the group can accept the immigrant. If the immigrant can be accepted, then the process ends. Otherwise, the next hexagon in the explored area is examined.

If an individual is unable to achieve its primary goal, and starting a group is the secondary goal, then the explored area is used to develop a list of cached ranges. Hexagons are added to the cached ranges in reverse chronological order, and as soon as a cached range is identified that meets the individual's target resource, it is taken. If no cached ranges ever attain the resource target then the process continues for each hexagon in the explored area. Then the best cached range overall will be taken, assuming it meets the minimum overall range resource criteria.

In all cases, as individuals explore, they constantly evaluate whether their primary goal can be met. As soon as this goal can be achieved, the exploration stops. If the goal cannot be met, the exploration will continue until the maximum explored area has been reached, or the individual dies due to an encounter with a barrier, or as a result of mortality spatial data. When the goal is to start a new group, then exploration only stops if 100% of the individual's resource target can be met. Otherwise exploration continues until the maximum explored area has been reached. Then the collection of cached ranges is examined to see if a sub-optimal range might be formed. In contrast, when the primary goal is to join a group, each target group is only examined once. If the group's range can supply the indicated percentage of the collective resource needs (which will be sub-optimal for Resource Threshold values less than 100), then it is joined. But unlike range initiation, the decision to settle into sub-optimal conditions is made during the initial encounter with the group.

If the primary goal cannot be met, but a secondary goal has been established, then the list of already explored hexagons is used to evaluate the possibility of achieving the secondary goal. If the secondary goal also cannot be met, or if there is no fall-back goal, then one of two things happen. If another dispersal can be taken, then the current explored area is abandoned and a new dispersal/ exploration cycle is initiated. If another dispersal/exploration cycle can not be taken, then the searcher remains a floater with an updated explored area.

When movement is set up to use multiple dispersal / exploration pairs, then decisions to start or join a group are based strictly on the current explored area. These decisions will not be based on any previous explorations that may have taken place. Further, when multiple dispersal/exploration cycles are being performed, the dispersal path will automatically avoid hexagons that were included in the most recent exploration. This should force dispersers to move away from a portion of the landscape that was just explored.

The division of ranges by range barriers is addressed during the range caching process that takes place within exploration. So bisected ranges will never be evaluated for suitability.

The Movement Event Global Tab

The Movement event's Global tab is used for specifying a population and set of trait combinations, movement barriers, and a mortality layer. It is also used to specify the movement strategy, number of explorations to be taken, and the order in which individuals are selected. At least one trait must be selected. Only individuals of that population who have the indicated traits will be affected by the movement event. An image of the Global tab is provided below.

Connectivity Test A | Movement (All Floaters)

Name: All Floaters

Global | Dispersal | Exploration | Output | Description

Population: Probes

Barriers: >>> None Selected <<<

☐ Mortality: >>> None Selected <<<

Strategy: Dispersal Then Exploration

[This movement strategy only operates on floaters.](#)

Maximum Number of Explorations: 1

☐ Give Movers Priority By Rank

Traits:

- ☒ Stages
- ☐ Status

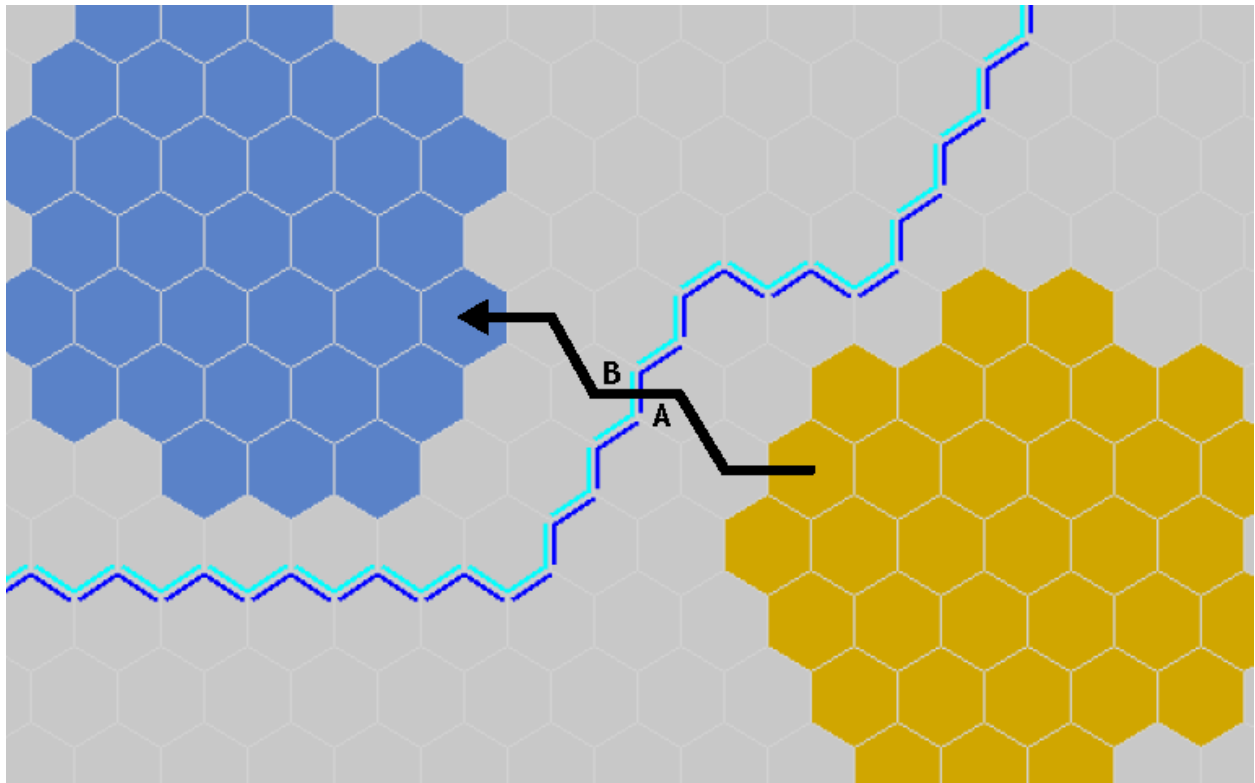
Combinations:

- ☒ Juvenile
- ☒ Sub-Adult
- ☒ Adult

Recover Close

If barrier spatial data is specified, then it will be applied to both the dispersal and exploration processes. HexSim barriers have probabilities for mortality, deflection, and transmission. Together these probabilities must sum to 1.0. All HexSim barriers have

two sides, which allows users to base a barrier's impact on the direction from which it is encountered. Barriers associated with a hexagon represent the **cost of exit** from that hexagon. For example, in the figure below, a blue barrier edge is encountered upon exit from hexagon A. In the same step, a cyan barrier edge is crossed upon entry into hexagon B. In this situation, the blue barrier edge will impact the disperser. There will be no consequence of crossing the cyan edge. Had the individual moved the other direction, then the cyan edge would have influenced it instead. This distinction can be ignored if both edges are assigned identical properties.



The same conventions hold true during exploration. Individual exploration steps involve moving from a previously visited site into an unexplored neighbor. The direction of travel is thus from the explored to the unexplored site, and this defines which side of a barrier is going to impact the individual. The patterns of exploration can result in multiple barrier crossings. Users should keep this in mind if barriers with some mortality are used in conjunction with exploration. Barriers can be made to impact dispersal but not exploration by using two Movement events. The first would use barriers with a movement strategy of Dispersal Only. The second would use a movement strategy of Exploration Only, but would not employ barriers.

HexSim can simulate a per-step probability of mortality during movement. If used, this will also impact both dispersal and exploration. This feature is used by clicking in the Mortality check-box, and then selecting mortality spatial data. The scores in the mortality spatial data define the probability of death associated with moving into each hexagon. For this reason, mortality spatial data will have to be carefully prepared. The

Mortality check-box, which is unchecked by default, is used as a safeguard to keep users mistakenly implementing per-step mortality.

In dispersal, users are allowed to set up repulsion so that some hexagons are never entered. This will not, however effect exploration. For example, the repulsion setting may ensure that dispersers never enter hexagons scored zero. But the same individuals will be free to explore zero-valued hexagons because exploration is not affected by repulsion and attraction. If a Movement event employs mortality spatial data, then it will become possible for individuals to explore and die in hexagons that, because of repulsion, they could not disperse into. This could appear very confusing. It is easy to make zero-valued (or hexagons with any value) inaccessible to both dispersal and exploration using barriers or an exclusion layer. If this is done, then there can be no apparent inconsistencies between dispersal and exploration.

HexSim provides six different movement strategies. These include Dispersal, Exploration, Dispersal Then Exploration, Exploration Then Dispersal, Translocate Then Explore, and Construct Explored Areas. The first four strategies dictate that either dispersal, exploration, or both dispersal and exploration will be performed. In the latter case, two strategies are provided so that users can control which process will come first.

Translocate Then Explore is a strategy that can be used when it is desirable to simulate the relocation of individuals without utilizing the step by step dispersal mechanism. Translocation is a process whereby individuals are moved directly from one hexagon to some other (possibly distant) hexagon, with no possibly adverse consequences. Translocation is always followed by exploration to ensure that these individuals have an opportunity to construct a range or floater allocation in the new location. When the Translocate Then Explore strategy is selected, users must also specify Translocation Spatial Data. Each individual undergoing translocation will select one non-zero hexagon at random from the translocation spatial data. Then that individual will be relocated to the selected hexagon. The Translocate Then Explore strategy will be useful for simulation some human activities (e.g. moving a tortoise away from roads) and for migrations that users do not want to simulate explicitly. Translocation is the only instance in HexSim where an individual is allowed to move directly from one hexagon to another non-adjacent hexagon.

Construct Explored Areas is the only movement strategy that operates on both group members and floaters. The other strategies act strictly on floaters. The Construct Explored Areas strategy forces individuals to perform a new exploration, but with the only goal being to build a new explored area. Group members will retain their group status and range data. Floaters will construct a new floater allocation (the floater analogy to a group's range) after the new explored area has been constructed. The size of the explored area that is constructed will be equal to the Maximum Explored Area parameter.

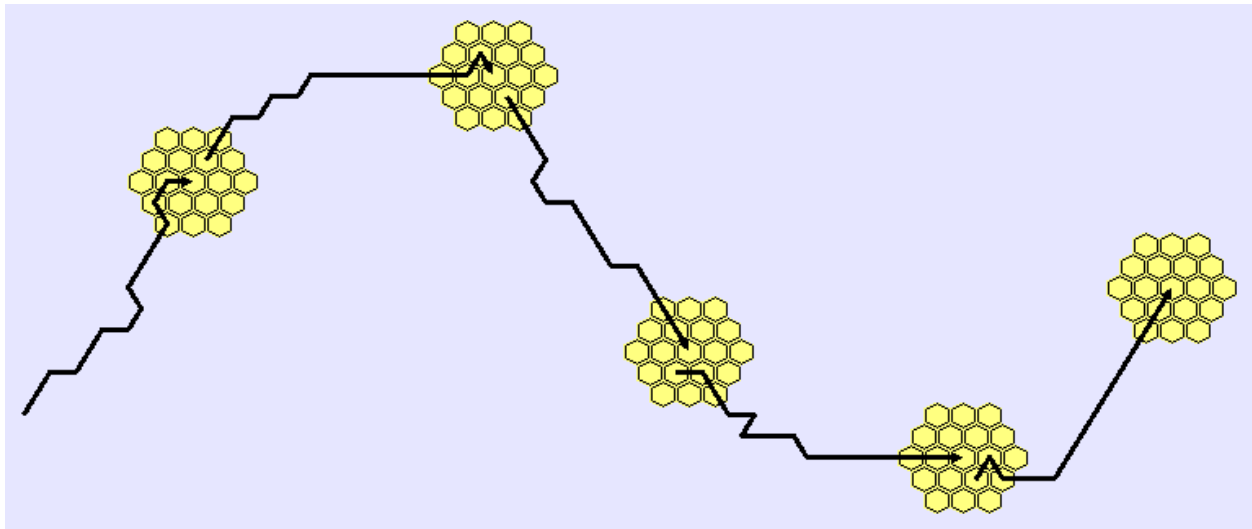
The Construct Explored Areas strategy will often be used prior to an Interaction event. Group members explored areas are automatically set equal to the range whenever a group is created or joined. But if a group member wanted to acquire prey items from a larger spatial area, but to do so without abandoning its range, then the Construct Explored Areas strategy is the mechanism that would make this possible. The Movement strategy would build a new (typically larger) explored area, which the individual could then use in an Interaction event to locate prey. Its group and range resources would remain unchanged.

HexSim has two updater function that can account for resource competition. These updater functions divide resources up among individuals (both within a group, and between populations) when their explored areas overlap. To make this possible, each population can form a per-hexagon tally of the number of individuals including the hexagon in their explored area. This is referred to as the Explored Counts tally, and it is only performed when a Movement event uses the Construct Explored Areas strategy. HexSim uses the Explored Counts to correctly divide up resources among each individual who shares a specific hexagon. As more individuals add a specific hexagon to their explored areas, the larger the tally for that hexagon becomes, and the less resources any single individual will be able to draw from it. When the Construct Explored Areas movement strategy is used, a Clear Explored Counts check-box becomes visible. The Clear Explored Counts check-box, when checked, instructs HexSim to zero out the Explored Counts tally. In most cases, this should be done once per time step, in the first Movement event using the Construct Explored Areas movement strategy. When used this way, Movement events implementing the Construct Explored Areas strategy and the Clear Explored Counts check-box can maintain an Explored Counts tally that accurately records the per-hexagon population densities across a landscape.

As stated above, the Explored Counts tally is only updated when a Movement event uses the Construct Explored Areas strategy. This is because when groups are formed or joined, the group member's explored area is set equal to the range. It is only through the use of the Construct Explored Areas movement strategy that group member explored areas can be made to extend beyond their ranges. (This statement is not strictly true, since Range Dynamics events, and the expansion of ranges associated with immigration can both modify range size without altering explored areas. For the moment, these exceptions are being ignored.) Thus, if users want to simulate competition using group member explored areas that are not equivalent to their ranges, then they *must* utilize a Movement event with the Construct Explored Areas strategy. Otherwise, competition should be evaluated based on ranges, and doing this does not require use of the Explored Counts tally.

The Maximum Number of Explorations parameter is used to make individuals perform multiple dispersal / exploration cycles. HexSim enforces a convention that if exploration is used, then the movement event must end with an exploration. Otherwise individuals might be left with no way to start or join a group. This means that if the movement strategy is set to Exploration Then Dispersal, a minimum of two explorations must be

conducted. Using multiple dispersal / exploration pairs enables individuals to move across the landscape, search a limited area, and then if the goal cannot be met, simply disperse and explore again. The figure below illustrates this process. The dispersal paths are shown as black lines, and the explored hexagons are colored yellow. Each exploration attempts to achieve its specified goal using only the hexagons visited in that exploration cycle. Previous explorations are effectively forgotten once the decision is made to disperse again. One exception is that the dispersal process will avoid hexagons visited during the most recent exploration. This ensures that when an exploration goal cannot be met, the ensuing dispersal will move individuals away from the just-visited hexagons



The Give Movers Priority By Rank check-box gives users some control over the order in which individuals move. When this check-box is not checked, individuals are selected for movement in a random order. When this check-box is selected, individuals are moved in order from highest to lowest priority. Priorities are specified in the [Population Parameters Range Data Tab](#), or in an [Adjust Range Parameters](#) event. Within a priority class, individuals are selected randomly. As movement proceeds and individuals start or join groups, available resources can become more and more limited. Thus this feature gives users a way to impose a hierarchy onto the resource acquisition process. Priority ranks also control the distribution of range resources among group members.

The Movement Event Dispersal Tab

The Dispersal event moves floaters around a landscape. Group members do not disperse. To apply dispersal to a group member, it must first be made a floater, which is done using a [Floater Creation event](#). Dispersal Spatial Data specify the land cover information that will influence the dispersal process. This may be distinct from the range spatial data used by the exploration component of movement for range construction. In image of the Dispersal tab is provided below.

The screenshot shows the 'Connectivity Test A | Movement (All Floaters)' dialog box with the 'Dispersal' tab selected. The 'Name' field is 'All Floaters'. The 'Dispersal' tab is active, showing various parameters for dispersal. The 'Dispersal Spatial Data' is set to 'Connectivity Test A'. The 'Distribution of Path Lengths' is set to 'Uniform'. The 'Log Normal Parameters (in hexagons)' are set to 'Mean: 0.0000' and 'Standard Deviation: 0.0000'. The 'Path Length Bounds (in hexagons)' are set to 'Minimum: 10' and 'Maximum: 50'. The 'Mean Resource Quality' is set to '80.00000'. The 'Use Stopping Criteria' checkbox is checked. The 'Experienced Over Path Length' is set to '5'. The 'Beginning with Dispersal Step' is set to '2'. The 'Repulsion' parameters are set to 'Maximum: -3.00000' and 'Minimum: 10.00000'. The 'Auto-Correlation' parameters are set to 'Minimum: 50.00000' and 'Maximum: 100.00000'. The 'Attraction' parameters are set to 'Minimum: 50.00000' and 'Maximum: 100.00000'. The 'Percent Auto-Correlation' is set to '75.0000' and the 'Trend Period' is set to '10'. The 'Use Affinity' dropdown is set to '>>> None Selected <<<'. The 'Recover' and 'Close' buttons are at the bottom right.

Name		Dispersal		Exploration		Output		Description	
Dispersal Spatial Data: Connectivity Test A									
Distribution of Path Lengths: Uniform									
Log Normal Parameters (in hexagons)		Mean		Standard Deviation					
		0.0000		0.0000					
Path Length Bounds (in hexagons)		Minimum		Maximum					
		10		50					
		Mean Resource Quality		80.00000					
<input checked="" type="checkbox"/> Use Stopping Criteria		Experienced Over Path Length		5					
		Beginning with Dispersal Step		2					
Repulsion		Auto-Correlation		Attraction					
-3.00000		50.00000		100.00000					
Maximum		Minimum		Maximum					
10.00000		100.00000		100.00000					
Percent Auto-Correlation		75.0000		Trend Period		10			
Use Affinity: >>> None Selected <<<									
Recover Close									

Each disperser will be assigned a path length. Path lengths are the number of steps that the disperser will move, and two distributions, Uniform and Lognormal, are available for

drawing path lengths. Uniform implies that each disperser will select its path length from a uniform distribution. When uniform is used, a path length minimum and maximum must be specified. The minimum and maximum path lengths may be set equal. The Lognormal option indicates that dispersal path lengths will be drawn from a lognormal distribution. The lognormal is characterized by a mean and a standard deviation. In addition, users must provide minimum and maximum path length parameters. If HexSim draws a lognormal random variant that falls outside of these bounds, the value will be rejected and another draw made.

HexSim obtains lognormally distributed random variates by transforming normally distributed ones. The transformation from normal to lognormal involves the fact that $Y = e^X$ will be lognormally distributed with mean M_L and standard deviation σ_L if X is normally distributed with mean M_N and standard deviation σ_N given by

$$M_N = \ln(M_L) - \frac{1}{2} \ln\left(\frac{V_L}{M_L^2} + 1\right), \text{ and } \sigma_N = \sqrt{\ln\left(\frac{\sigma_L^2}{M_L^2} + 1\right)}$$

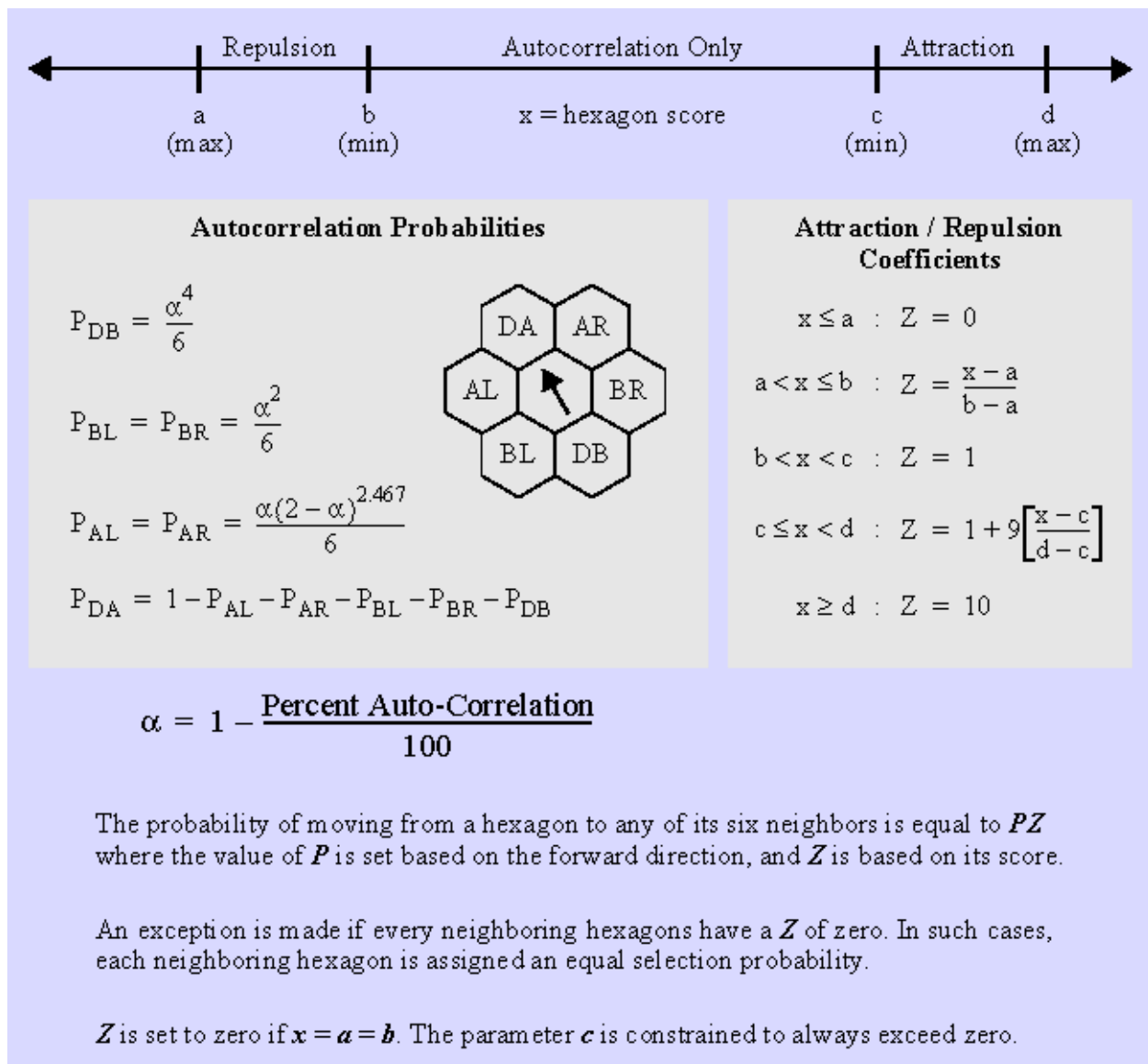
The user supplies M_L and σ_L , and HexSim uses the equations above to obtain M_N and σ_N . A normally distributed random variate is obtained from $N(M_N, \sigma_N)$, and then exponentiated to obtain a lognormally distributed random variate with the desired mean and standard deviation.

The path length parameters are all specified in hexagons. The path length defines how many steps (from one hexagon to a neighbor) each individual will move. There are conditions, described below, under which an individual will stop its dispersal prior to moving the full path length.

Dispersal stopping criteria make it possible for individuals to quit moving before going the full path length. The dispersal process is informed by resource quality, and it is not aware of resource availability. The stopping criteria thus specify a mean resource quality threshold that, if encountered over any portion of the path, will halt dispersal. As each individual moves, HexSim keeps track of the mean hexagon quality experienced over the indicated number of steps (the value 5 in the figure above). If this mean value reaches the quality threshold (the value 80.0 in the figure above), then the dispersal is stopped. A third parameter (with a value of 2 in the figure above) allows users to determine at what dispersal step this process will commence. The intent is that users may control both the quality and amount of resource that must be encountered to stop the dispersal process. The use of a running mean can prevent dispersers from stopping in tiny islands of good habitat. A consequence of this scheme is that many dispersers may elect to stop in the same general location, and their collective resource targets may exceed the available resource. However, this will be sorted out during the exploration

process, and those individuals who cannot meet their exploration goals may be allowed to disperse again to another location.

HexSim allows users to define repulsion and attraction parameters to guide dispersal paths away from some landscape elements, and towards others. There is also an auto-correlation setting that makes dispersal paths more or less random. In the absence of repulsion and attraction, zero auto-correlation produces a true random walk. At the other extreme, 100% auto-correlation will result in straight-line movement trajectories. However, repulsion, attraction, and auto-correlation all work together to determine the dispersal path characteristics.



The figure above illustrates how repulsion, attraction, and auto-correlation, work together to control dispersal behavior. (Grid edges, barriers and excluded areas will also influence dispersal, but we will ignore these complications for the moment.) Each

hexagon has six neighbors and taking a single dispersal step involves selecting one neighbor and moving to it. Each neighbor is assigned a value PZ , where P is set based on auto-correlation, and Z reflects any repulsion or attraction. Hexagons can be either repulsive, neutral, or attractive. The auto-correlation probabilities P will be values between zero and one. Z will lie between zero and one if a hexagon is repulsive, will be exactly one if it is neutral, and will be greater than one if a hexagon is attractive. Once PZ has been computed for each neighbor, the values are normalized by dividing each by the sum. Thus, each hexagon is ultimately assigned a probability that captures both auto-correlation and the influence of attraction or repulsion. To select a neighbor, a random variate is drawn from $U[0, 1]$ and compared to the individual neighbor probabilities (the normalized PZ values). The larger a neighbor's probability, the greater the likelihood that it will be selected.

Auto-correlation is implemented by assigning higher probabilities to directions that represent forward movement. HexSim therefore constantly tracks the direction of past movements. The left gray box in the figure above illustrates the past direction with a black arrow, and the abbreviations DA, AL, AR, BL, BR, and DB are used to label the neighbors that are directly ahead, ahead left, ahead right, behind left, behind right, and directly behind. These labels are placed relative to the forward direction. HexSim uses a Trend Period parameter to better define the forward direction. The Trend Period is a number of steps selected by the user, and HexSim tracks the forward direction for each step in this period. For example, if the Trend Period is set to 5, then the forward direction will be stored for each of the last 5 steps. The forward direction actually used to label the six neighbors (that is, locate DA, DB, etc) will be the direction that occurs most frequently over the Trend Period. The use of Trend Periods adds a kind of momentum to auto-correlated dispersal paths.

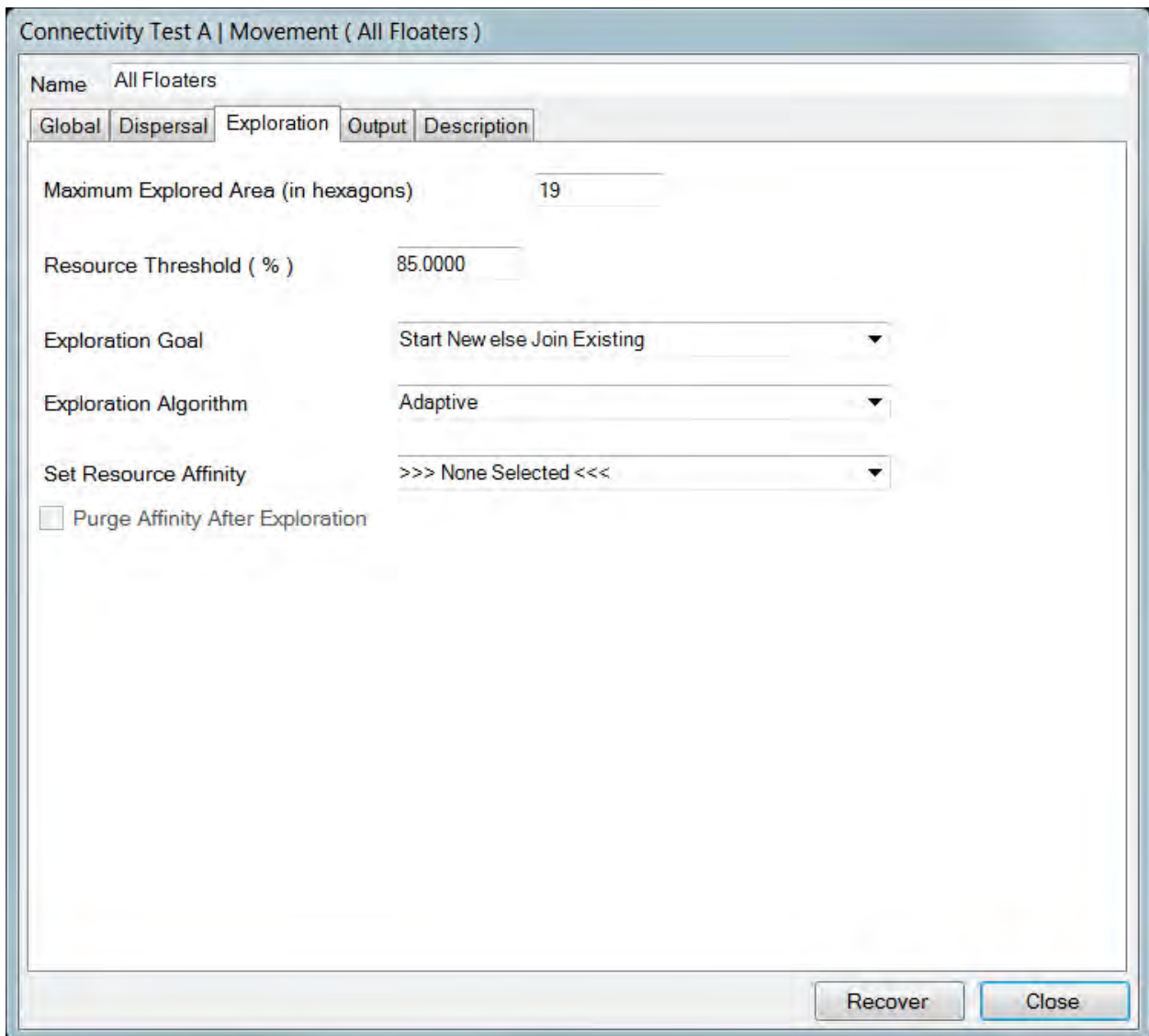
Once the DA, AL, AR, BL, BR, and DB labels are attached to the appropriate neighbors, then each is assigned an auto-correlation probability, P . The equations used to assign P values are listed in the figure above. The six autocorrelation probabilities (P_{DA} , P_{AL} , P_{AR} , P_{BL} , P_{BR} , P_{DB}) are all smoothly varying and sum to 1.0. The expressions for P_{AL} and P_{AR} are designed so $P_{DA} = P_{AL} + P_{AR}$ when the Percent Autocorrelation parameter is set to 50%.

Repulsion and attraction produce a coefficient (Z) which is multiplied by the auto-correlation probability, P . The right gray box in the figure above shows how this value Z is computed. The number line at the top of the figure illustrates the separate attraction and repulsion zones, which may not overlap. A single hexagon can be either repulsive ($x < b$), attractive ($x > c$), or neither (neutral). For hexagons with a score less than the maximum repulsion, Z is fixed at zero. As the hexagon's score increases from the maximum to minimum repulsion values, Z increases linearly from zero to one. Z remains at one until the hexagon's score increases to the minimum attraction value. When the hexagon's score increases from the minimum to maximum attraction value, Z again increases linearly. The maximum value for Z is 10, which is applied to hexagons with scores meeting or exceeding the attraction maximum.

The dispersal affinity selector is used to direct movement paths towards predefined affinity sites. Users may select any natal, reproduction, resource, or group movement affinity that has been set up in the population parameters. Affinity will be ignored if an individual has no stored value for the selected affinity type. Affinity works by hijacking dispersal's record of the forward direction. Instead of drawing the forward direction from the recent movement history, as specified by the Trend Period, affinity sets the forward direction towards the neighboring hexagon closest to the affinity target. For this reason, the Trend Period selector becomes inactive when dispersal affinity is used. The rest of the process described above proceeds unchanged. The higher the auto-correlation, the more focused the movement paths will be towards the affinity target. Repulsive and attractive hexagons may divert individuals away from an affinity target. If the affinity target is reached, then dispersal will immediately stop for that individual.

The Movement Event Exploration Tab

The exploration process involves an intensive search for resources. Exploration is the only process through which groups and ranges can be initiated. The exploration process is also responsible for creating individual explored areas, and floater allocations (the floater analog to a group's range). And image of the Exploration tab is provided below.



Connectivity Test A | Movement (All Floaters)

Name All Floaters

Global Dispersal **Exploration** Output Description

Maximum Explored Area (in hexagons) 19

Resource Threshold (%) 85.0000

Exploration Goal Start New else Join Existing ▼

Exploration Algorithm Adaptive ▼

Set Resource Affinity >>> None Selected <<< ▼

☐ Purge Affinity After Exploration

Recover Close

Users specify a Maximum Explored Area, in hexagons. It is also necessary to set an exploration goal. Some goals have primary and secondary components. In these cases, if the primary goal is listed first, and if it cannot be met, then an attempt is made to attain the secondary goal. For the most part, goals involve starting a new group or joining an

existing group. Four of HexSim's six goals are built up from simple combinations of these: Start a New Group, Start New else Join Existing, Join an Existing Group, and Join Existing else Start New.

The two additional goals are slight variations on those above. They are Start New else Join Existing New, and Join Existing New else Start New. These both involve the notion of an "existing new" group. An existing new group is simply a group that was formed during the current movement event. These two goals are useful when forming multi-individual groups that are made up exclusively of the individuals acted on by the current Movement event.

The exploration process can be conducted using one of three exploration algorithms. These algorithms define the methods used to select which hexagon to explore. The starting point of each exploration is the individual's location, which is typically the end point of dispersal. As hexagons are entered, they are added to the current explored area. Only immediate neighbors of the set of explored hexagons can be entered, but this collection grows every time a new hexagon is visited.

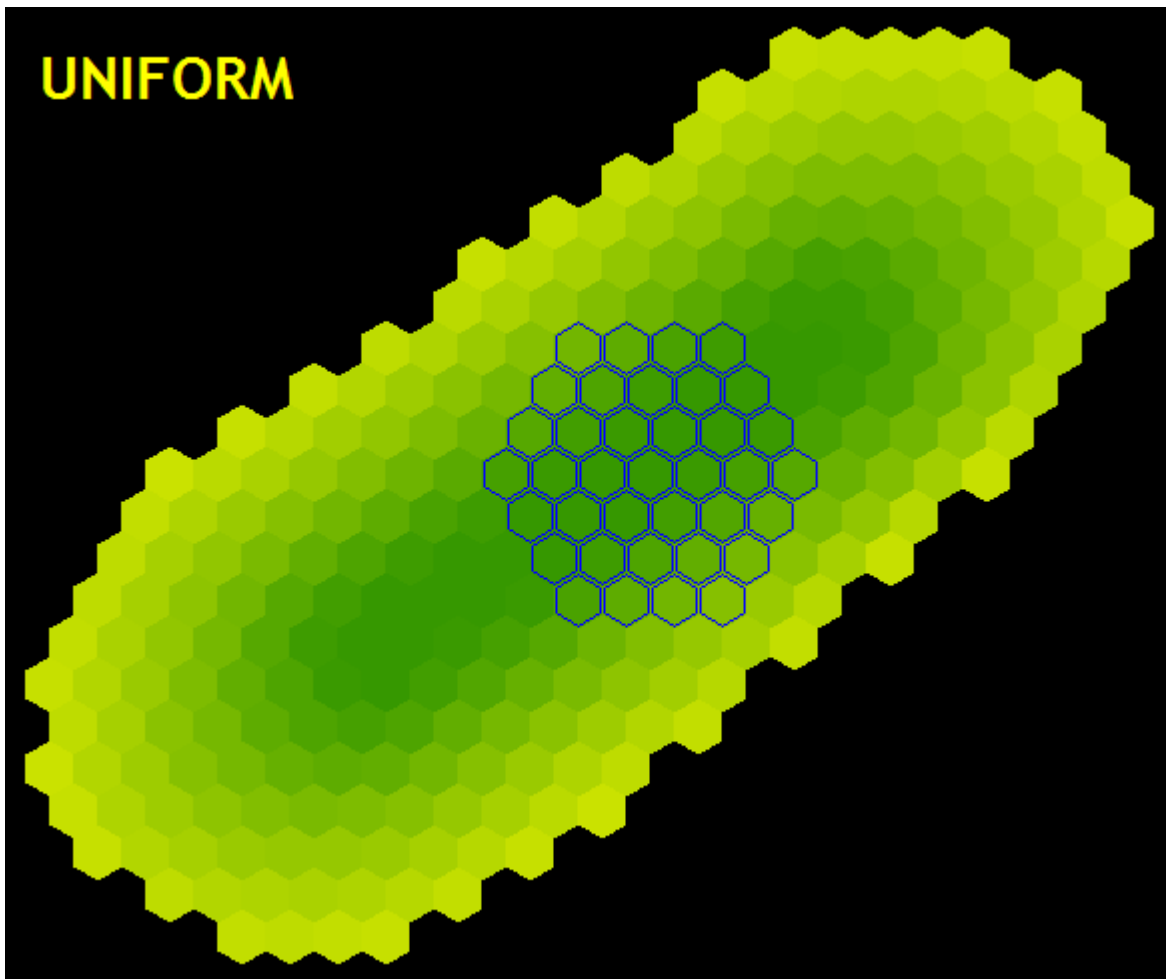
Under the Uniform algorithm, the closest unexplored neighbor to the exploration starting point will always be selected. Ties are settled randomly. This algorithm tends to produce roughly circular explored areas. Still, the landscape edges, excluded areas, and barriers must be respected. So the ultimate search area may not be a simple set of concentric rings.

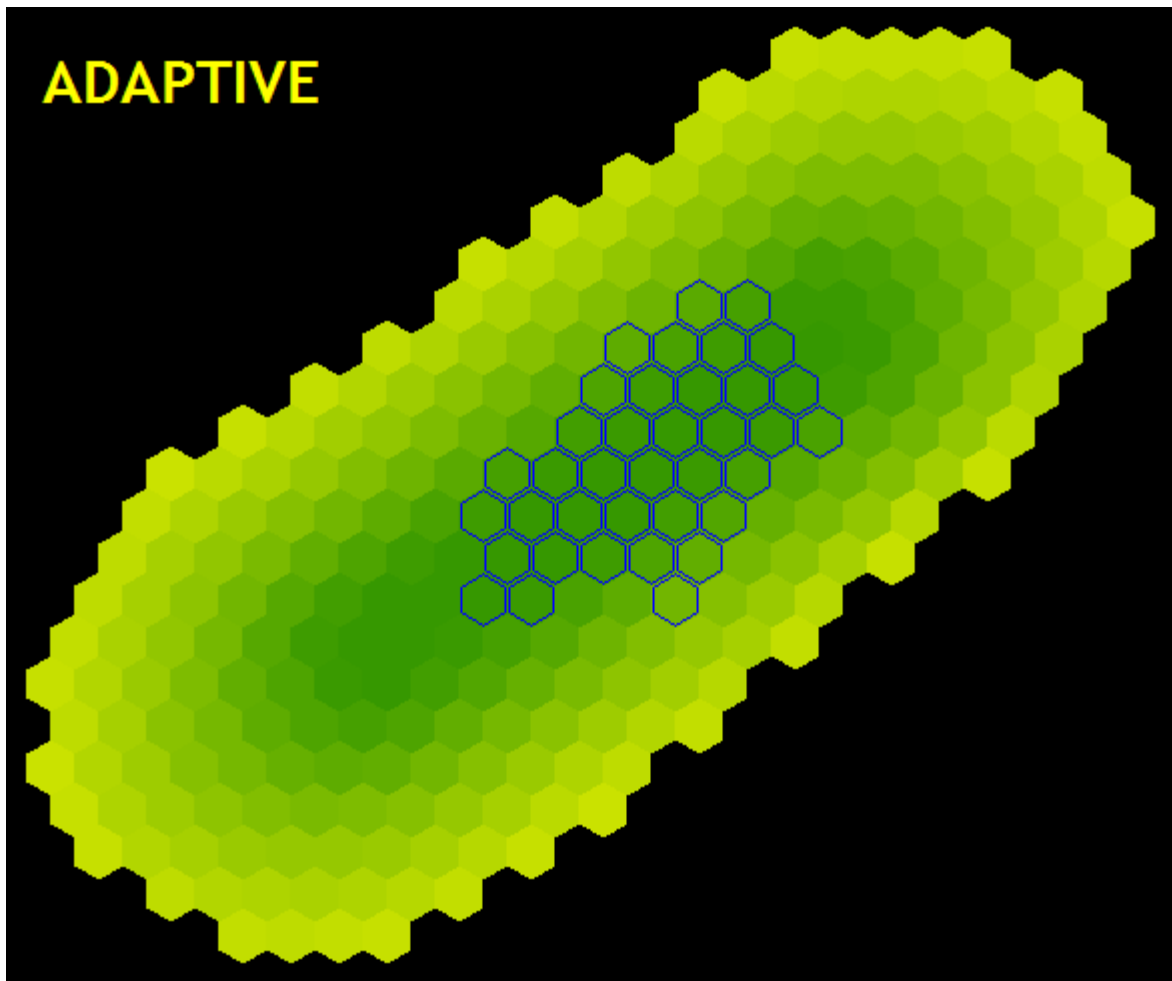
The Greedy algorithm sorts the list of unexplored neighbors and always selects the best of these as the next site to visit. Ties are settled randomly. Again, landscape boundaries, excluded areas, and barriers are all taken into consideration. For both the Uniform and Greedy strategies, the process continues until a range that meets the individual's target resource has been found, or the Maximum Search Area has been reached. In either case, the entire explored area becomes the new range, as long as it meets the minimum resource criteria for ranges.

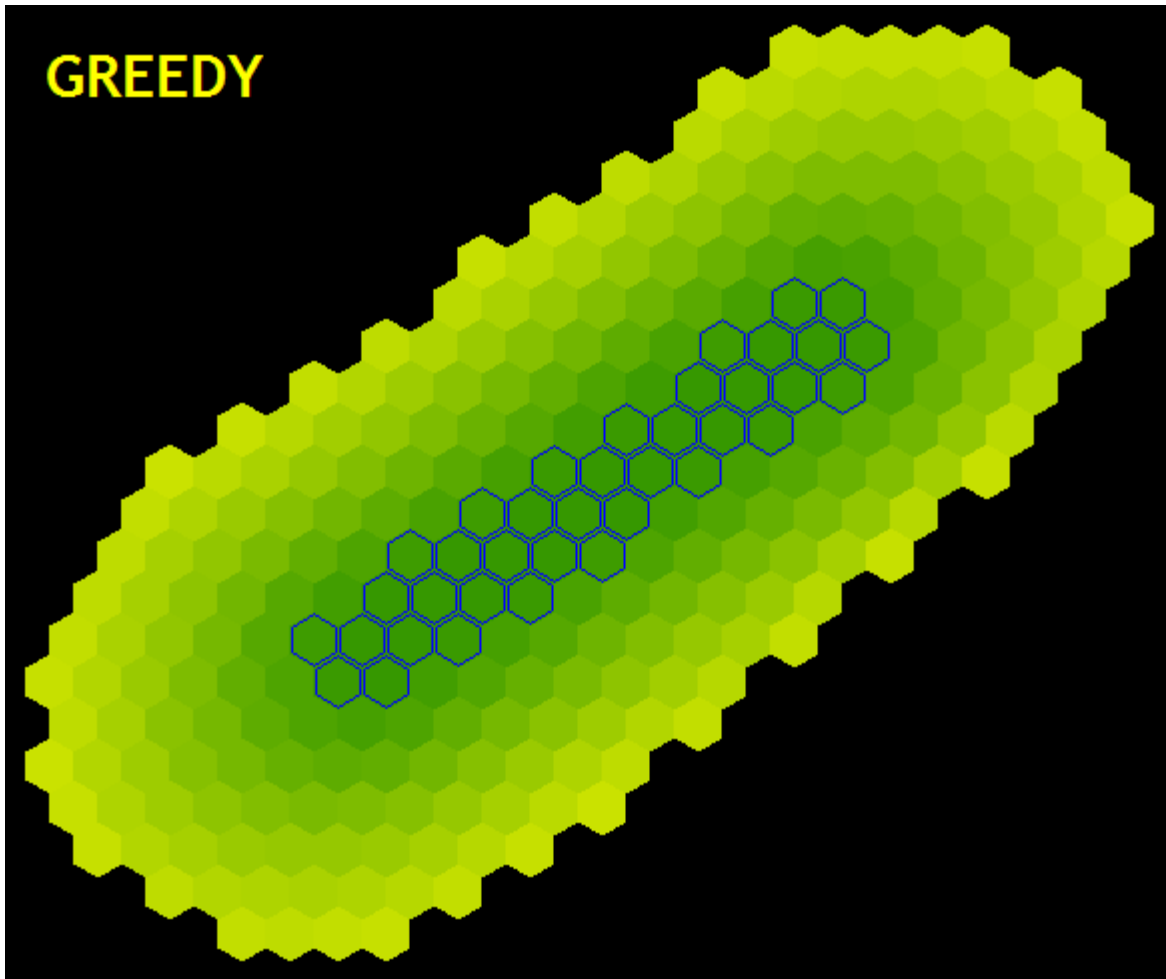
The Adaptive exploration strategy is a bit more complex. When picking a new site to visit, the Adaptive strategy first picks a seed site from the collection of hexagons that have already been explored. This seed hexagon is selected probabilistically, based on quality. Then each of the seed hexagon's neighbors is considered for exploration. These neighbors are evaluated based both on their quality and on the number of previously explored neighbors they have. The reason for including the number of previously explored neighbors in the evaluation is that it helps keep the ranges compact. The number of explored neighbors is simply used as a coefficient for the hexagon score. Unexplored neighbors with 1, 2, 3, 4, 5, and 6 explored neighbors are assigned coefficients of 1.0, 1.2, 1.4, 1.6, 1.8, and 2.0, respectively. These coefficients are multiplied by the unexplored neighbors scores. These products are then rescaled so that they sum to 1.0, and used as a probability. The choice of a neighbor to visit is then made by drawing a random variate from $U[0, 1]$.

The Greedy and Uniform algorithms represent limiting cases that simulate complete knowledge of one's immediate surroundings (greedy), or a total lack of such knowledge (uniform). The Adaptive strategy, which simulates limited knowledge of one's surroundings, will often be more biologically realistic. The Adaptive algorithm typically produces lower quality explored areas than the Greedy algorithm, but it is also less constrained, and so it may sometimes branch out and discover good habitat clusters that the Greedy algorithm will miss.

The figures below illustrate the differences between the three algorithms. In the figures, the hexagon scores trend from low at the outside edge to high in the center. The lower hexagon scores are more yellow in color, and higher hexagon scores are more green. The hexagons outlined in blue illustrate the explored area that was created using each of the three algorithms. All three explored areas contain equal numbers of hexagons. The Uniform strategy is deterministic. The Greedy strategy is also deterministic, except that ties will be settled randomly. The Adaptive strategy is highly stochastic.





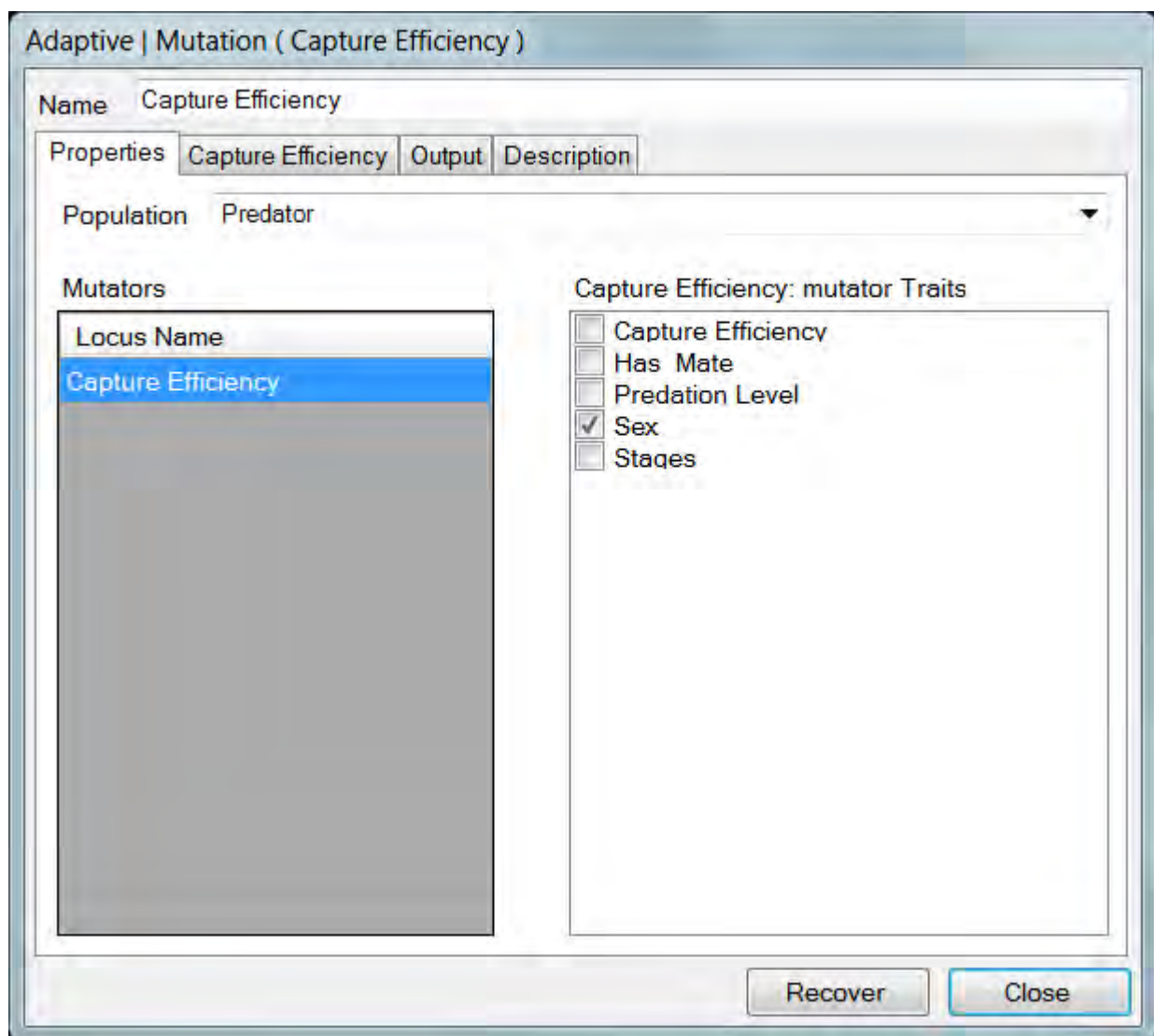


The Set Resource Affinity menu allows users to select a previously defined resource affinity. If an affinity is supplied here, it will be used to store a record of ranges whose quality exceeds the threshold value assigned to that affinity register.

The Purge Affinity After Exploration check-box, when checked, makes individuals purge stale affinity records from their affinity registers. Affinity records can become stale when landscape change reduces resource quality. If an individual returns to a previously visited site only to find that it no longer meets the threshold associated with the affinity register, then that affinity record will be removed.

The Mutation Event

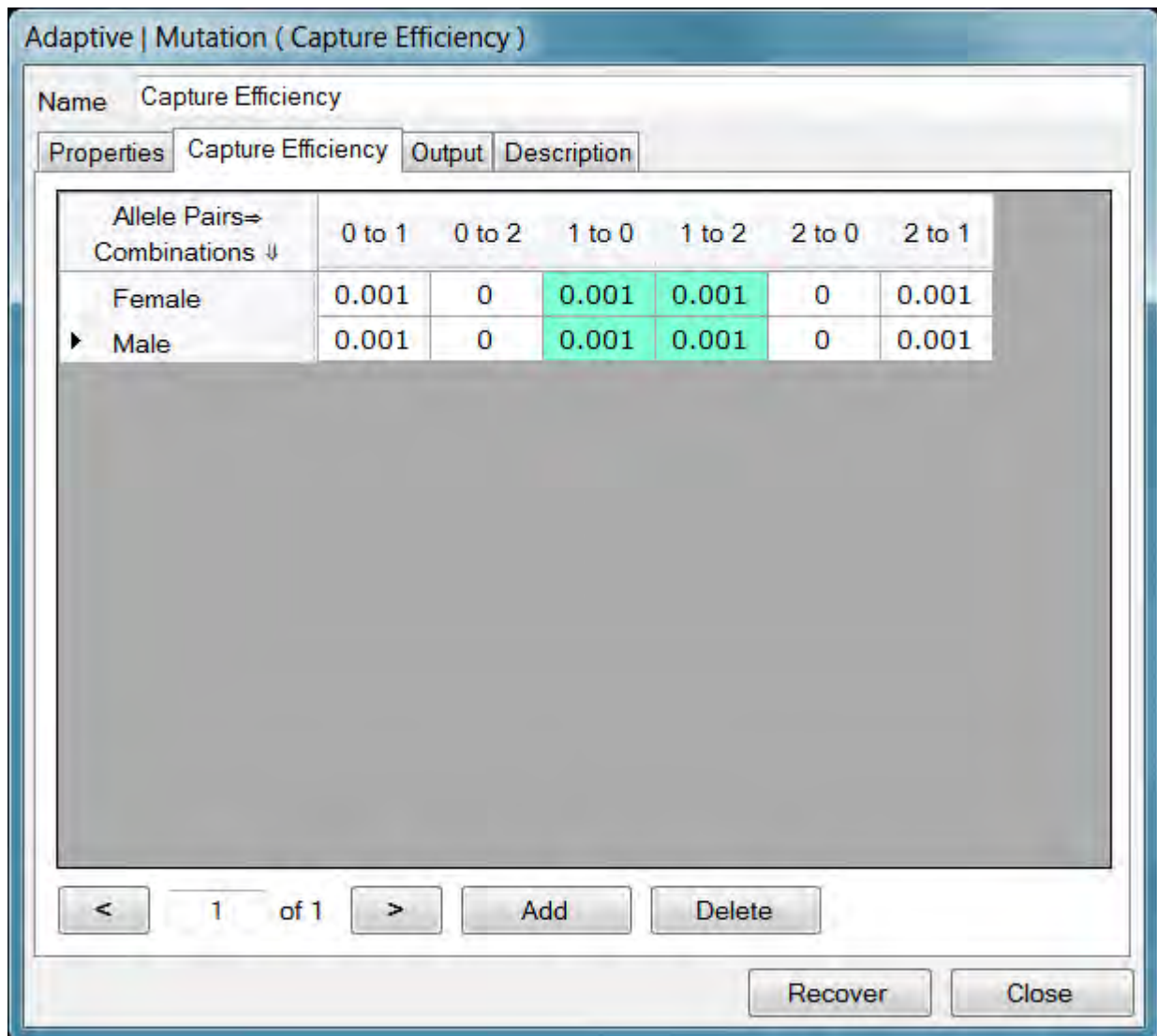
HexSim's Mutation event can change which alleles are present at one or more loci. But the Mutation event cannot create alleles that did not previously exist in the population's genome. Alleles intended for use in simulating mutation may be left out of the initial population's genotypes so they can emerge only as the result of a mutation event. Allele definition is performed using the [Traits tab](#) of the Population Parameters window, and is discussed further in the section on [HexSim Genetics](#). An image of the Mutation event is provided below.



Mutation events are conceptually simple, but their use may involve the specification of a large number of probabilities. Each mutation event can operate on one or more loci.

Users select one or more loci to as subjects for mutation. For each loci, one or more mutation traits are also selected. Typically, the mutation traits will be either probabilistic or accumulated. For example, exposure to a mutagenic toxin might be captured in an accumulated trait, which is in turn used to stratify mutation probabilities in a Mutation event.

The loci and mutation traits are specified in the Mutation event's Properties tab. When a loci is selected, its mutation probabilities are displayed in a dedicated tab, which is assigned the name of the currently selected locus. We will refer to this tab as the *Mutation Probabilities tab*, and to the table it holds as the *Mutation Probabilities table*. An image of a Mutation Probabilities tab is provided below.



The mutation probabilities tab is assigned the name of the locus that it controls. The mutation stratification trait combinations are assigned to rows in the mutation probabilities tab. The columns are used to enumerate the possible allele transitions.

HexSim does not include columns corresponding to transitions with identical from and to traits, for example 0 to 0. The probability that an allele will not mutate (e.g. a 0 to 0 transition) is simply 1.0 minus the sum of the probabilities that it will mutate. For this reason, the allele mutation probabilities for a given row in the mutation probabilities table may sum to less than 1.0. HexSim uses color to distinguish the columns in the mutation probabilities table that correspond to transitions from a single allele (see figure above).

A context menu is provided with each mutation rates tab. This context menu allows users to set rates to zero, import and export mutation rates, and to change the trait combination permutation ordering.

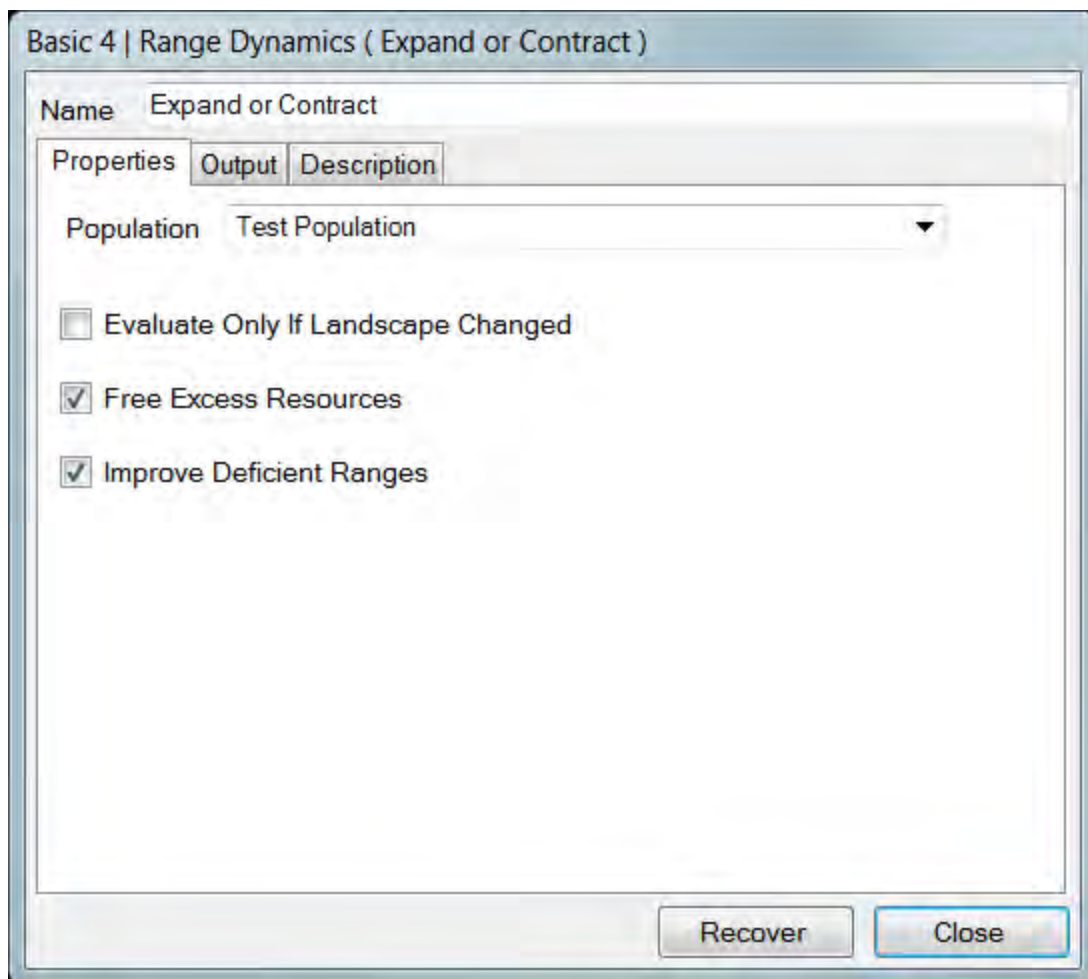
- The Zero All Data option has a sub-menu that allows users to choose a row, column, or matrix.
- The Import Values and Export Values context menu options read in, or write out the tables of mutation rates. Simple text files are used for this purpose. The import and export functions will work with single data tables, or collections of tables.
- The Order Rows by Selected Traits context menu option will change the way that the traits values are sequenced, if more than a single Stratification Trait has been selected in the Properties tab. These changes are permanent, but can be easily altered by using the tool again. Only the traits combinations are reordered, so be careful using this feature. If mutation rates have already been entered, reordering the traits will change alter the pairings between rates and traits.

Changing the selection of stratification traits will clear the data from the mutation probabilities tab. The Add button must be used to create a new mutation probabilities table when the mutation probabilities tab is empty. Additional stochasticity may be built into HexSim Mutation events through the use of multiple mutation probability tables. If multiple tables are supplied, then one will be drawn from the collection each time the Mutation event fires. Specific tables are drawn from the collection randomly, or in order, based on the stochasticity model (Random versus Cyclic) selected by the user. See the discussion under Simulation Parameters in the section on the [Scenario Tabs](#).

When the Mutation event runs, individual genotypes may be changed. The consequence of an altered genotype can be a change in trait values for one or more genetic traits that are linked to the mutated alleles. If other life history events such as survival or reproduction depend in some way on these genetic traits, then the Mutation event can have important consequences for the population.

The Range Dynamics Event

The Range Dynamics event will make it possible for groups to modify their ranges. Mortality or emigration can cause ranges to exceed the resource needs of the remaining group members. Similarly, reproduction and immigration can cause range resources to fall below the group's needs. Finally, landscape change can either increase or decrease range resources. Range Dynamics events provide groups with a way to respond to these changing resource conditions. An image of the Range Dynamics event is displayed below.



Range Dynamics events have three parameterization options, and users may select any combination of these. Range Dynamics may be triggered only during time steps in which the landscape has changed. Also, Range Dynamics may be instructed to free excess resources, or to improve deficient ranges. If excess resources are freed, the

range will shrink in size. If resources are added, then ranges will typically increase in size.

When a deficient range is expanded, the group will be given access to all of the existing range hexagons plus every available hexagon that is immediately adjacent to the existing range. If a superior range can be constructed from these hexagons, then the group will give up the old range and switch to the new one. If a superior range can not be constructed, then the original range will be retained. Any new hexagons added to the range are also added to the group's explored area.

When freeing excess resources, group explored areas are left unchanged.

If landscape change makes an entire range invalid, then one hexagon will be selected at random from the old range. The entire range will then be collapsed down to that single hexagon.

Range dynamics can cause ranges to cross movement barriers. If such barriers should be respected by Range Dynamics events, then they must be used as Range Barriers in the [Population Parameters Range Data tab](#), or in an [Adjust Range Parameters](#) event.

The Reproduction Event

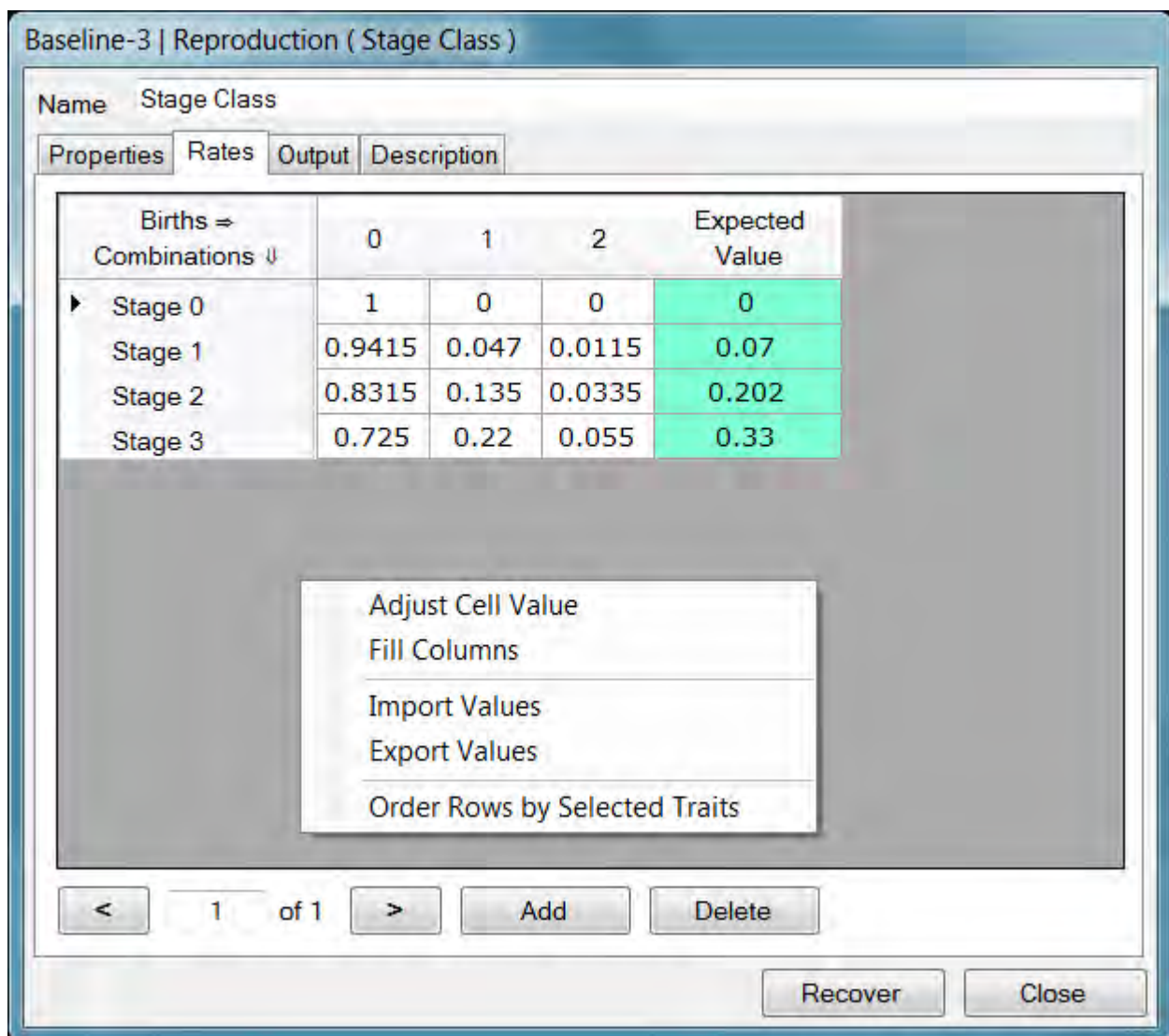
HexSim's Reproduction event achieves considerable flexibility using few parameters. Users select a population and one or more traits for use in stratifying the reproductive rates. It is then necessary to specify a maximum number of births. This parameter will typically represent a biological upper limit. Reproduction events can set both Natal and Reproduction Affinities if called for in a simulation. Affinities are described in detail in the section titled [Population Parameters Affinities tab](#). Reproduction events can also perform temporary pairings between individuals in two-sex simulations. An image of the Reproduction event is displayed below.

The screenshot shows a window titled "Baseline-3 | Reproduction (Stage Class)". Inside, there are several tabs: "Properties", "Rates", "Output", and "Description". The "Properties" tab is active. It contains the following fields and controls:

- Name:** Stage Class
- Population:** A dropdown menu currently showing "Female Spotted Owls".
- Maximum Number of Births:** A text input field containing the number "2".
- Set Natal Affinity:** A dropdown menu showing ">>> None Selected <<<".
- Set Reproduction Affinity:** A dropdown menu showing ">>> None Selected <<<".
- Use Mate Selection:** An unchecked checkbox.
- Traits:** A list of traits with checkboxes:
 - ☐ Barred Owl Prese
 - ☐ Demographic Stu
 - ☐ Modeling Region
 - ☐ Resource Class
 - ☒ Stage Class
 - ☐ Territory Status

At the bottom right of the window are two buttons: "Recover" and "Close".

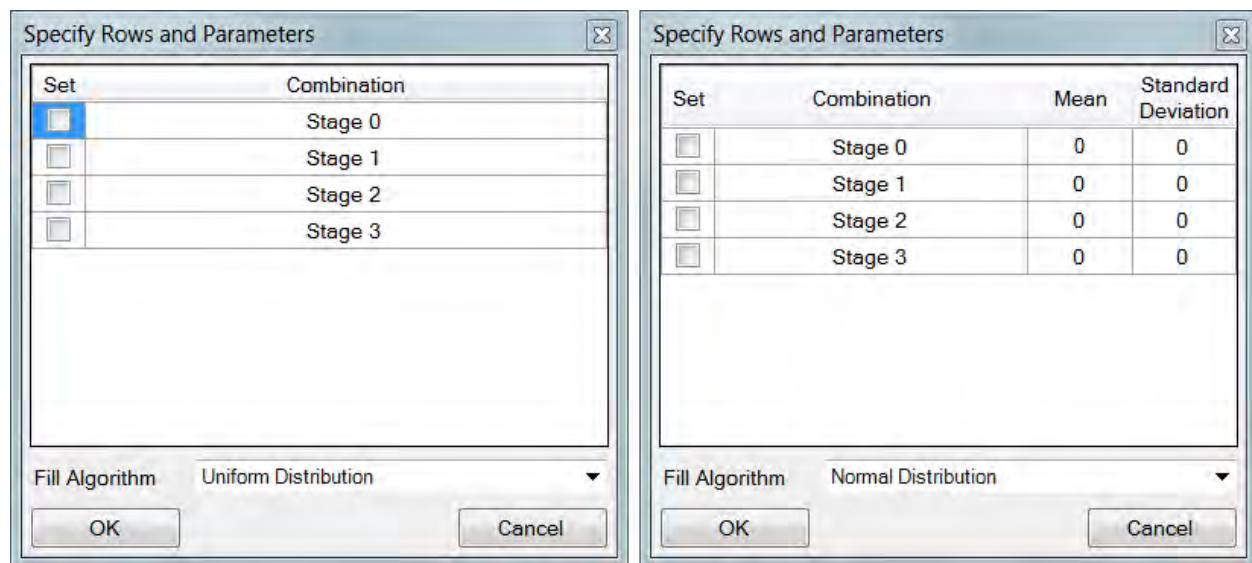
Once the Maximum Number of Births has been specified, and one or more traits have been selected, the reproductive rates can be added to the Rates tab. Initially, the Rates tab will be empty, but a data table will appear if the Add button is clicked. The rows of this table are the combinations of the trait values selected in the Properties tab. The columns of this table enumerate the possible numbers of births. The column headings will range from 0 to the value assigned to the Maximum Number of Births parameter in the Properties tab. Each cell in this table must be assigned a probability, and every row must sum to exactly 1.0. If a trait combination should not reproduce, then a 1 should be placed in the zero-births column (which will necessitate that zeros be placed in the other columns, so that the row total sums to 1.0). For each trait combination, the expected value is reported in the right-most column. An image of the Reproduction event's Rates tab is shown in the figure below.



A context menu is provided with the Rates tab (see figure above). This context menu allows users to automatically adjust cell values, automatically fill entire rows, import and export reproductive data, and change the trait combination permutation ordering.

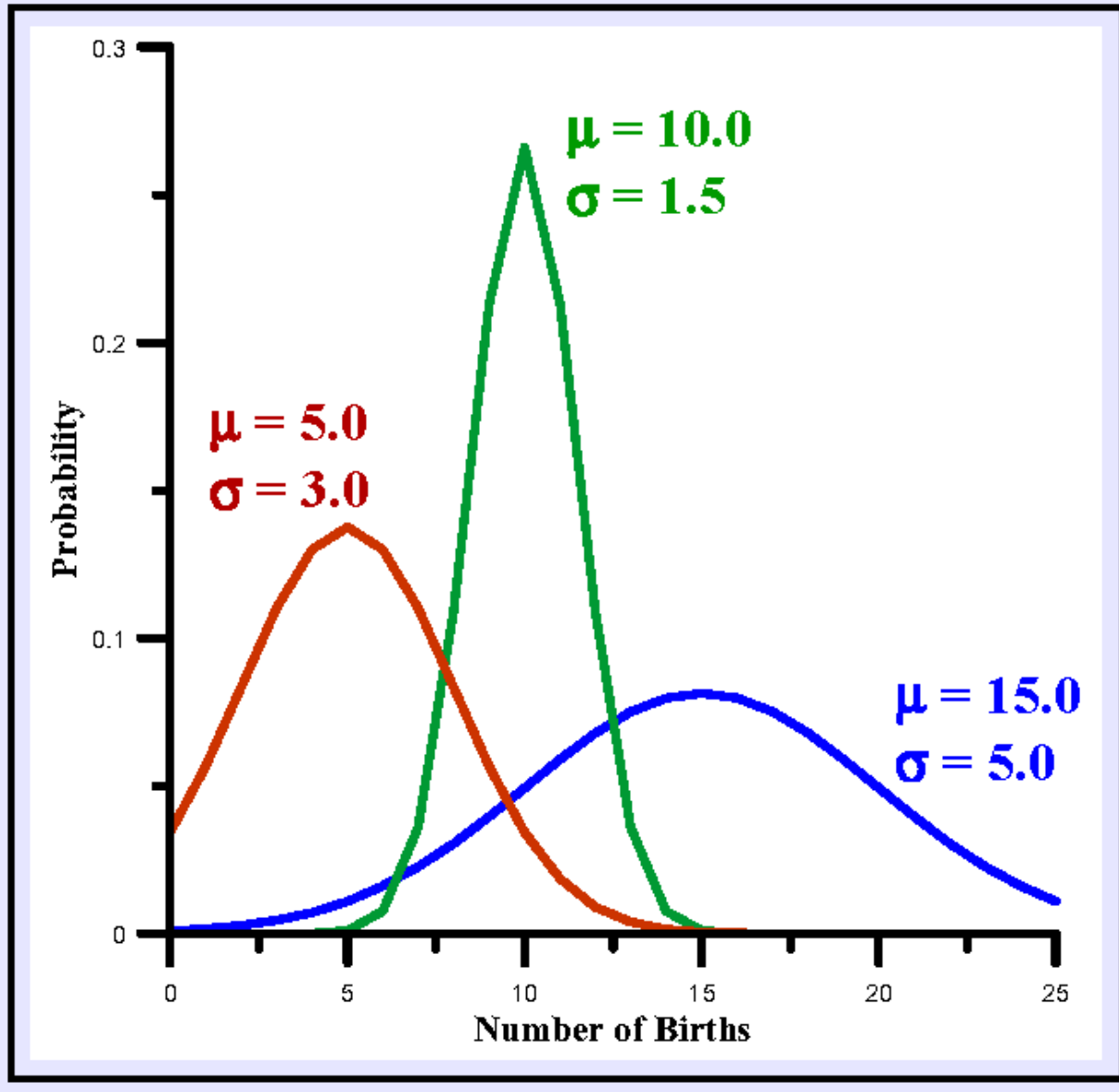
- The Adjust Cell value context menu option will set the selected cell equal to 1.0 minus the sum of the other cells in that row. This can be a convenient time-saver.
- The Import Values and Export Values context menu options read in, or write out the rate table data. Simple text files are used for this purpose. The import and export functions will work with single rate tables, or collections of tables.
- The Order Rows by Selected Traits context menu option will change the way that the traits values are sequenced, if more than a single trait has been selected in the Properties tab. These changes are permanent, but can be easily altered by using the tool again. Only the traits combinations are reordered, so be careful using this feature. If reproductive rates have already been entered, reordering the traits will change alter the pairings between rates and traits.
- The Fill Columns context menu option provides an automated mechanism for populating the data table rows.

The Fill Columns utility can be useful if the Maximum Number of Births parameter is large, since then the reproduction data table would be difficult to populate one cell at a time. The Fill Columns tool uses one of two Fill Algorithms to supply probabilities to every column in one or more rows. Users select the rows to be modified by the tool. If the Fill Algorithm is set to Uniform Distribution, then each of the N columns in a row will all be assigned a probability of $1/N$. This has limited utility, but requires no additional parameters. If the Fill Algorithm is set to Normal Distribution, then a normalized Gaussian curve is used to assign relative probabilities to the different outcomes (columns). This distribution requires setting a mean and a standard deviation for each selected row. Images of the Fill Columns dialog windows are shown in the figure below.



When the Normal Distribution Fill Algorithm is used with Fill Columns, each of the possible numbers of births is assigned a probability given in the expression for P , below. In this expression, x represents a specific number of births, μ represents the mean, and σ represents the standard deviation. The denominator simply rescales the P values so that they sum to 1.0. A plot is also shown below that illustrates how the probabilities assigned to the outcomes change with μ and σ .

$$P = \left[e^{-\frac{(x-\mu^2)}{2\sigma^2}} \right] \div \left[\sum_x e^{-\frac{(x-\mu^2)}{2\sigma^2}} \right]$$

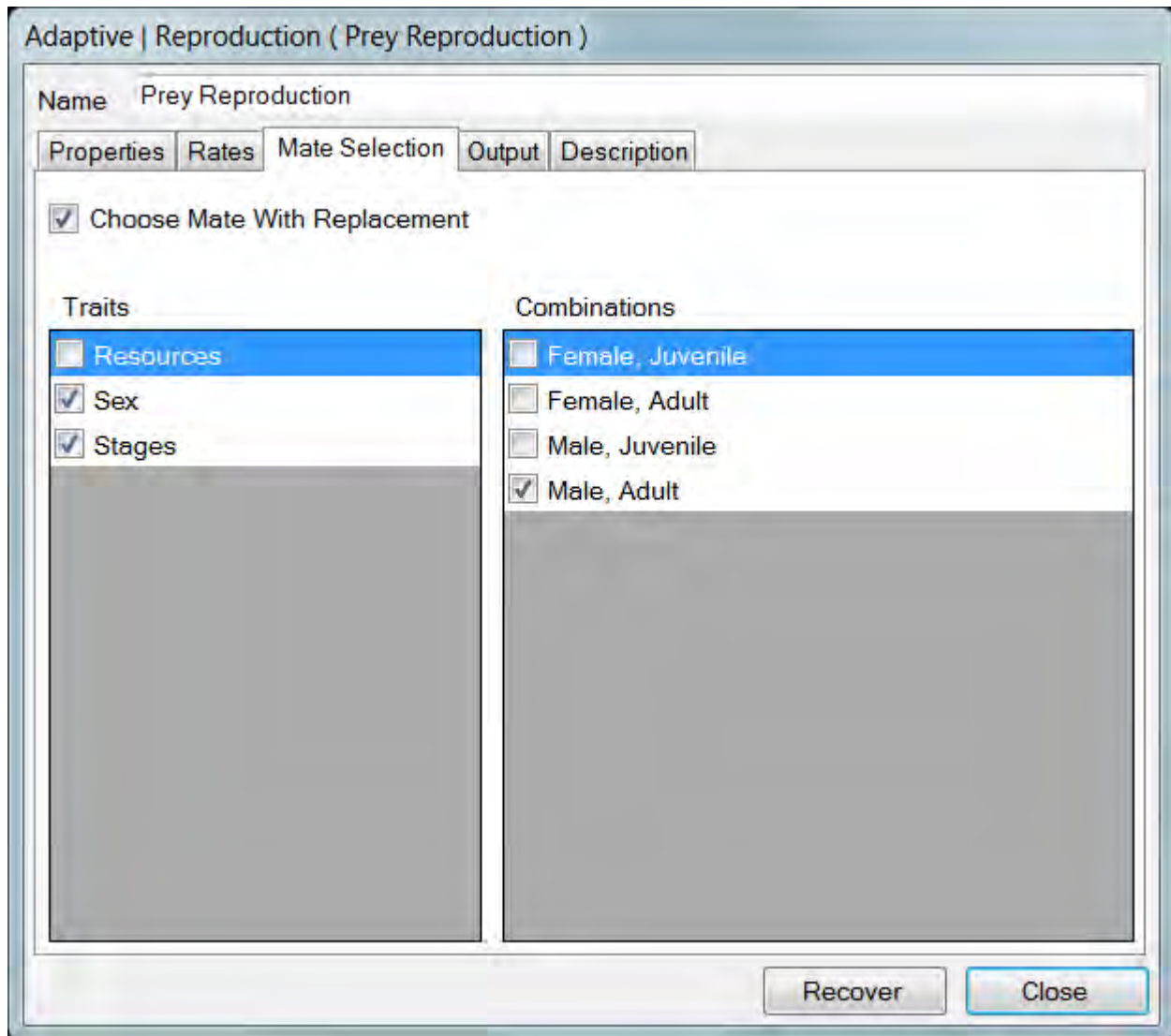


The collection of buttons at the bottom of the Reproduction event's Rates tab can be used to navigate between multiple rate tables. An image of these buttons is shown below. Users can also add and delete these tables. Whenever the choice of traits in the

Properties tab changes, the reproduction rate data will be lost. It is then necessary to either add a new table, or import one that has been stored in a file. Multiple rate tables can be used to simulate environmental stochasticity. If multiple rate tables exist, then one member of the collection will be selected each time the event runs. For more information on simulating environmental stochasticity, see the discussion under Simulation Parameters in the section on [Scenario Tabs](#).



The Use Mate Selection check-box makes it possible to temporarily pair up group members for the purposes of reproduction. When this check-box is checked, a Mate Selection tab is created (see the figure below). The mate selection tab is used to set the criteria that individuals must have in order to qualify as a mate. This simply involves specifying traits and trait combinations. Group members that possess the selected trait combinations qualify as mates. These mates should be male individuals. The females are those individuals who have been assigned non-zero reproductive rates in the rates tab. When the Mate Selection tab is in use, females will only be allowed to reproduce if they are assigned a suitable mate. User's may choose to form exclusive pairings, or to select mates with replacement. If the Choose Mate With Replacement check-box (see figure below) is checked, then every suitable mate in the group will be available to every breeding female. If this check-box is not checked, then suitable mates may only pair with a single female. In either case, these pairings only last during for that instance of the Reproduction event.



If a Reproduction event does not use Mate Selection (the Use Mate Selection box is unchecked), then each individual will reproduce at the indicated rate. Even if the Use Mate Selection box is checked, unpaired individuals can reproduce if they possess one of the trait combinations selected in the Mate Selection tab. In this case, they simply qualify as their own mate. To limit reproduction to situations in which an actual mate is present, users must do one of two things:

1. Use Mate Selection, and specify trait combinations for the mate that distinguish it from the reproducing individual. For example, reproducing individuals could be restricted to those having a female trait value, and the set of candidate mates could be restricted to those having a male trait value.
2. Set up genetics, and use the Mate Accumulator tools. In this case, a trait must be used to identify whether or not a female is paired. The reproductive rates are then stratified by this trait, and unpaired individuals are assigned zero reproduction.

The Set Group Affinity Event

Animals that live in groups such as herds, packs, or flocks, often move in groups as well. The Set Group Affinity event is designed to make such movements possible in HexSim. Ordinarily, individuals disperse completely separately. Set Group Affinity operates on a group-by-group basis. What it does is to move a "proto-disperser" and record where it ended up. The proto-disperser can be instructed to look for resources sufficient to support a collection of individuals. Once the event ends, the proto-disperser is deleted. The location identified by the proto-disperser gets passed back to the group, and is then distributed to members when they leave the group. These members can then disperse towards the affinity site when they move. The Set Group Affinity event is a slightly restricted analog of a Movement event. An image of a Set Group Affinity event is shown below.

Group Movement | Set Group Affinity (Find New Group Location)

Name Find New Group Location

Global Dispersal Exploration Output Description

Population Critters ▼

Barriers >>> None Selected <<< ▼

Maximum Number of Explorations 1

Group Affinity Group Movement Affinity ▼

Recover Close

Most of the parameters in the Set Group Affinity event also appear in the Movement event. So for help with these parameters, please refer to the various sections of this document that describe Movement events. The exceptions are described below.

The Global tab of the Set Group Affinity event is mostly the same as the [Movement event's Global tab](#). The differences are that Set Group Affinity does not have mortality spatial data, a strategy selector, or a priority ranking check-box. The mortality spatial data and the priority ranking option simply have no purpose in group movement. The strategy used with group movement is always Dispersal Then Exploration.

The one element present in the Set Group Affinity Global tab that is not present in the Movement event is a Group Affinity selector. This tool lets users specify which group movement affinity variable the results should be stored in.

The only difference between the Set Group Affinity event's Dispersal tab and the [Movement event's Dispersal tab](#) is that Set Group Affinity cannot target a hexagon stored in an affinity record. Thus the Use Affinity selector is missing from the Set Group Affinity Dispersal tab.

The Exploration tab for the Set Group Affinity event allows users to set the Maximum Explored Area and the Exploration Algorithm. Both of these parameters appear in the [Movement event's Exploration tab](#) as well. The Set Group Affinity event also has a Resource Target setting that does not appear in the Movement event. This is equivalent to the Resource Target parameters found in the Population Parameter's Range Data tab, but in this case it applies strictly to the proto-disperser. Users should typically set this Resource Target parameter high enough to meet the needs of a large group. The Set Group Affinity event does not have an Exploration Goal, and can not create resource affinity data.

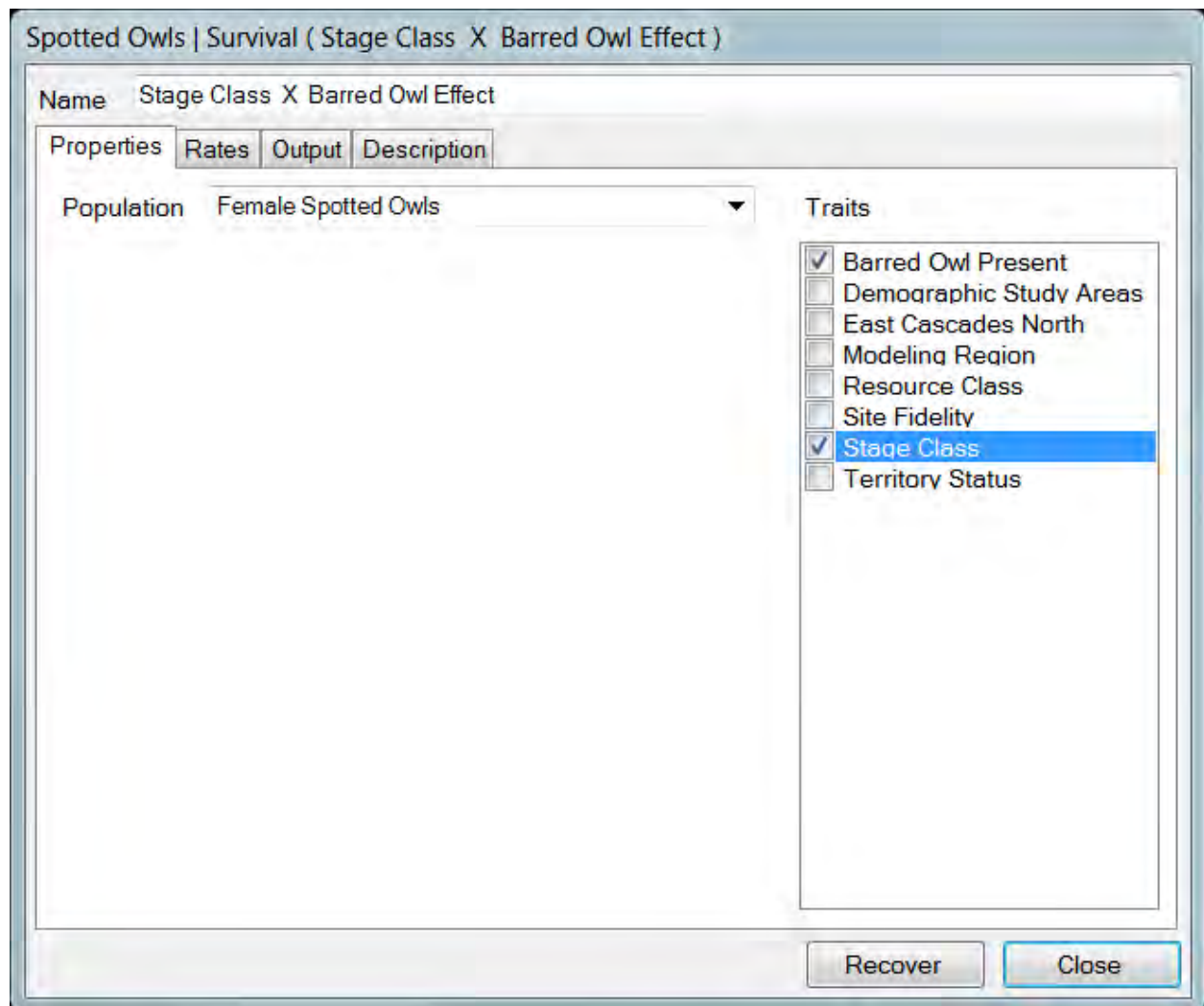
Set Group Affinity events create a dummy individual (the "proto-disperser"), and allow this individual to move and search for a new range to build. Once a suitable range has been found, the center of the range is identified and used as the affinity site. That affinity value is then assigned to each member of the group. This is repeated for each group. If no suitable range can be found then the center of the best substandard range is used as the affinity site. If no legitimate substandard ranges can be identified, then Set Group Affinity does nothing. That is, the affinity register for that group is left empty.

Set Group Affinity events are usually followed by [Floater Creation](#) events, and then Movement events. Set Group Affinity will assign a group-affinity to each group member. Floater creation is necessary since only floaters will move. Movement will need to be set up so that the dispersal part of the event uses the group affinity. Set Group Affinity sets a group-affinity, but when group members become floaters they take a copy of any group affinity data with them. This is true even for individuals who joined the group after the affinity data was created. This makes it possible for these floaters to disperse towards an affinity site assigned to their previous group. The affinity data that Set Group Affinity assigns to a group will stay with the group unless it is overwritten. Affinity data can only be used by the dispersal component of a Movement event, so it has no value to group members, since only floaters can disperse.

Sometimes a floater will disperse towards a group affinity site, but fail to reach it, or end up unable to join the new group. Such an individual will continue to retain the group affinity data. That affinity data can become stale with time, and to control for this the Set Group Affinity event will zero the group affinity registers for all floaters. Note however that multiple group affinity registers may be created for a population. Only the affinity register selected in the Set Group Affinity event will be cleared.

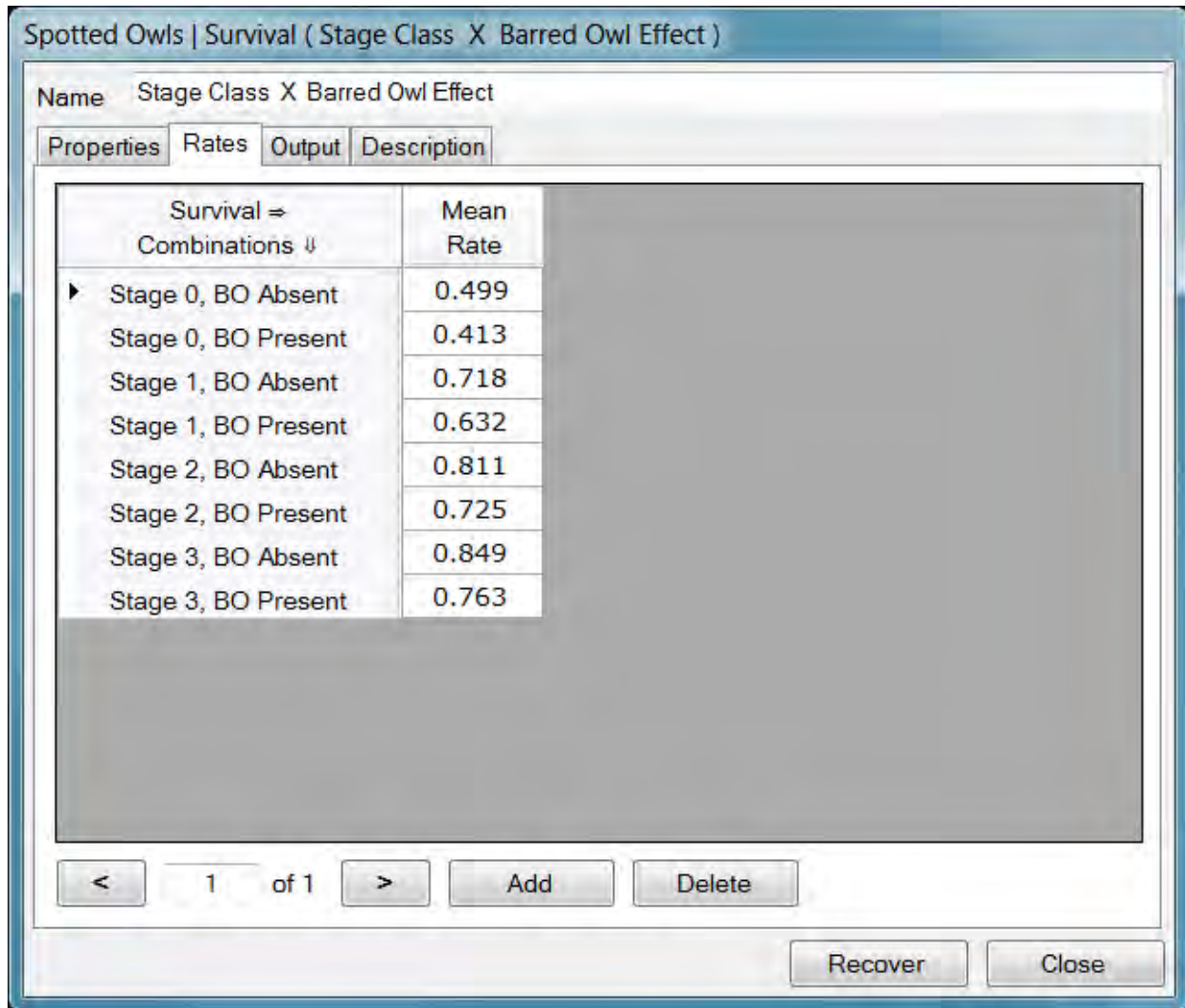
The Survival Event

HexSim's Survival event is simple and intuitive. Users need only select a population and one or more trait values. Once this has been done, a series of survival probabilities can be entered into the Rates tab. The values placed into the Rates tab are probabilities that an individual will survive. Probabilities of mortality are equal to 1.0 minus the survival rates. Survival rates are stratified by the permutations of trait values for the selected traits. An image of the Survival event is shown below.



When one or more traits have been selected, users can place survival probabilities in the Rates tab. Anytime the trait selection is changed, the data on survival rates will be lost. In this case, users should use the Add button to create a new data table. Alternatively, survival rate data that has previously been saved to the disk can be

imported into the event using the Rates tab's context menu. The Survival event's Rates tab is shown below.



The collection of buttons at the bottom of the Survival event's Rates tab can be used to navigate between multiple rate tables. Users can also add and delete these tables. Multiple rate tables can be used to simulate environmental stochasticity. If multiple rate tables exist, then one member of the collection will be selected each time the event runs. For more information on simulating environmental stochasticity, see the discussion under Simulation Parameters in the section on [Scenario Tabs](#).

A context menu provided with the Rates tab allows users to import and export survival data, and change the trait combination permutation ordering. The Import Values and Export Values context menu options read in, or write out the rate table data. Simple text files are used for this purpose. The import and export functions will work with single rate tables, or collections of tables.

The Order Rows by Selected Traits context menu option will change the way that the traits values are sequenced, if more than a single trait has been selected in the Properties tab. These changes are permanent, but can be easily altered by using the tool again. Only the traits combinations are reordered, so be careful using this feature. If survival rates have already been entered, reordering the traits will change alter the pairings between rates and traits.

The Transition Event

Transition events are used to change the values of probabilistic traits. Users select a population, a Transition Trait and one or more Stratification Traits. The Transition Trait is the probabilistic trait whose values are going to be altered. The Stratification Traits make it possible to vary the transition probabilities based on other data stored with an individual. Probabilistic, accumulated, and genetic traits can all be used for stratification. This flexibility makes the Transition event quite powerful.

Suppose for example that disease status (healthy, infected, sick) has been captured in a probabilistic trait. Assume as well that a measure of fitness has been captured in an accumulated trait. The transition between disease classes will depend on what class an individual is in (e.g. a healthy individual must become infected before it can get sick). But these transitions might depend on fitness as well, since a highly fit individual may recover more rapidly than one who is poorly fit. To capture this dynamic in a Transition event, users would use the disease status trait as the Transition trait, and would use both traits as stratification traits. For additional realism, a second accumulated trait tracking exposure to the disease could be added, and used to further stratify the Transition event.

An image of the Transition event illustrating this simplistic example is shown below.

Disease Introduction | Transition (Set Disease Status)

Name: Set Disease Status

Properties | **Transitions** | Description

Population: Test Population

Transition Trait: Disease Status

Stratification Traits

- ☒ Disease Status
- ☐ Disturbance Class
- ☒ Exposure Class
- ☒ Fitness Class
- ☐ Resource Class
- ☐ Stage Class

Recover Close

Once the Stratification traits have been selected, users can supply transition probabilities in the Transitions tab. Making changes to the Stratification traits will cause the Transitions tab data to be lost. In these cases, the Add button can be used to create a new data table, or data can be imported directly into the event.

The transition trait values will be represented as columns in the data table. The permutations of all stratification trait values will make up the rows. Each cell in the table must be assigned a probability, and the values on any given row must sum to exactly 1.0. Every individual in the population will be operated on by the event. Each individual's trait values will correspond to exactly one row in the data table. The probabilities on that row are then used to determine how the Transition Trait should be set. A probability of exactly 1 will ensure that this transition trait value will be assigned to all individuals. This approach (setting one cell in a row to 1) can be used to effectively turn off the Transition event for selected classes of individuals.

An image of the Transition event's Transitions tab is shown below.

Disease Introduction | Transition (Set Disease Status)

Name

Properties Transitions Description

Transition To ⇒ Combinations ↓	Healthy	Infected	Sick
▶ Healthy, Unexposed, Low Fitness	1	0	0
Healthy, Unexposed, Moderate Fitness	1	0	0
Healthy, Unexposed, High Fitness	1	0	0
Healthy, Exposed, Low Fitness	0	1	0
Healthy, Exposed, Moderate Fitness	0	1	0
Healthy, Exposed, High Fitness	0	1	0
Infected, Unexposed, Low Fitness	0	0	1
Infected, Unexposed, Moderate Fitness	0	0	1
Infected, Unexposed, High Fitness	0	0	1
Infected, Exposed, Low Fitness	0	0	1
Infected, Exposed, Moderate Fitness	0	0	1
Infected, Exposed, High Fitness	0	0	1
Sick, Unexposed, Low Fitness	0	0	1
Sick, Unexposed, Moderate Fitness	0.25	0	0.75
Sick, Unexposed, High Fitness	0.5	0	0.5
Sick, Exposed, Low Fitness	0	0	1
Sick, Exposed, Moderate Fitness	0.25	0	0.75
Sick, Exposed, High Fitness	0.5	0	0.5

< 1 of 1 > Add Delete

Recover Close

A context menu is provided with the Transitions tab. This context menu allows users to automatically adjust cell values, import and export transition data, and change the trait combination permutation ordering.

- The Adjust Cell value context menu option will set the selected cell equal to 1.0 minus the sum of the other cells in that row. This can be a convenient time-saver.

- The Import Values and Export Values context menu options read in, or write out the transition probabilities. Simple text files are used for this purpose. The import and export functions will work with single data tables, or collections of tables.
- The Order Rows by Selected Traits context menu option will change the way that the traits values are sequenced, if more than a single Stratification Trait has been selected in the Properties tab. These changes are permanent, but can be easily altered by using the tool again. Only the traits combinations are reordered, so be careful using this feature. If transition rates have already been entered, reordering the traits will change alter the pairings between rates and traits.

The collection of buttons at the bottom of the Transition event's Transitions tab can be used to navigate between multiple rate tables (see figure above). Users can also add and delete these tables. Whenever the choice of Transition Traits in the Properties tab changes, the transition probability tables will be lost. It is then necessary to either add a new table, or import one that has been stored in a file. Multiple rate tables can be used to simulate environmental stochasticity. If multiple rate tables exist, then one member of the collection will be selected each time the event runs. For more information on simulating environmental stochasticity, see the discussion under Simulation Parameters in the section on [Scenario Tabs](#).

Analysis

Analysis

[Census Files](#)

[Reports](#)

[Tallies](#)

[Display Spatial Data](#)

[The Simulation Viewer](#)

Census Files

HexSim's census files are the easiest outputs to generate and the simplest to work with. Every Census event will generate an output file, in a CSV (comma-separated variable) format. Users can place as many Census events into the life cycle as they want. The CSV output files generated by Census events are assigned a numeric suffix which links the file to the event. The first Census event in the life cycle will produce an output file with a suffix of 0. The second will produce a file with a suffix of 1, and so on. Census events are discussed in the section titled [The Census Event](#).

Census event output is useful for making plots of population size with time. However, population size is tracked in several different ways in these output files. There are columns for the total population size, the numbers of group members and floaters, and for the number of individual with each combination of the traits selected in the event's interface. Finally, an instantaneous measure of Lambda (the rate of population growth) is provided. The instantaneous lambda value for time step T is equal to $[\text{Population at T}] / [\text{Population at T-1}]$. This simple computation can be cumbersome to perform by hand, so its is generated as a convenience. The lambda value for time step zero is always set to one.

An image of a HexSim Census event output file, as displayed in a spreadsheet, is shown below.

HexSim User's Guide

Grapher - [Basic 4.0.csv]

File Edit View Format Data New Graph Tools Window Help

W 0 H 0 X 0 Y 0

Basic 4.0.csv

	A	B	C	D	E	F	G	H	I
	Run	Time Step	Population Size	Group Members	Floater	Lambda	Trait Index 0	Trait Index 1	Trait Index 2
1	1	0	500	0	500	1	0	0	500
2	1	1	600	597	3	1.2	239	0	361
3	1	2	568	568	0	0.946667	164	115	289
4	1	3	565	565	0	0.994718	183	84	298
5	1	4	573	573	0	1.014159	174	106	293
6	1	5	576	576	0	1.005236	188	87	301
7	1	6	564	564	0	0.979167	183	95	286
8	1	7	566	566	0	1.003546	172	103	291
9	1	8	576	576	0	1.017668	180	94	302
10	1	9	593	593	0	1.029514	203	89	301
11	1	10	592	592	0	0.998314	179	105	308
12	1	11	617	617	0	1.04223	207	99	311
13	1	12	605	604	1	0.980551	182	112	311
14	1	13	648	648	0	1.071074	226	99	323
15	1	14	627	627	0	0.967593	196	120	311
16	1	15	625	625	0	0.99681	190	113	322
17	1	16	637	637	0	1.0192	203	105	329
18	1	17	607	607	0	0.952904	164	113	330
19	1	18	620	620	0	1.021417	204	85	331
20	1	19	602	601	1	0.970968	180	116	306
21	1	20	624	624	0	1.036545	208	111	305
22	1	21	612	612	0	0.980769	201	110	301
23	1	22	587	587	0	0.95915	174	109	304
24	1	23	607	607	0	1.034072	192	98	317

Reports

HexSim reports are accessed through the Scenario drop-down menu. Reports provide a variety of information about what actually happened in HexSim a simulation. Reports are generated entirely from log files, and this is done after a simulation has completed. It is not necessary to anticipate in advance what reports will be required. Every report can be run as long as every logging option was set active for every event in the life cycle. Logging is controlled through the event Output tabs, and is always on by default. Turning logging off may speed up a simulation slightly, and it will definitely reduce the size of log files. But HexSim's ability to generate reports and other products from a log file will be limited if some simulation data has not been logged in the file.

At present, there are 12 different HexSim Reports. The reports all produce CSV files intended to be read into a spreadsheet. The Dispersal Steps, Dispersal Paths, and Projection Matrix reports produce data tallies, so users must supply initial and final time steps. The tallies are then gathered only for the specified period. The remaining 9 reports are stratified by replicate and time step, so there is no need to specify initial and final time steps. If there are multiple populations in a simulation, then a CSV report file will be generated for each one.

Each of the HexSim reports is described below.

Births

The Births report produces a record for every birth that took place during a simulation. This data is gathered exclusively from Reproduction events, but multiple such events may exist within a life cycle.

The last column in a Births report is the Mate ID. If the mate ID is not known, then a value of zero is written there. If the "Use Mate Selection" check-box, in the Reproduction event, is checked, and if HexSim is set up correctly so that this function can work, then mates will be assigned. As a result, the Mate ID column of the Births report will hold non-zero values.

The other way to assign mates is to create a mate accumulator and use an Interaction event to assign values to this accumulator. But this does not provide the Reproduction event with enough information that it can assign mate IDs to newborn log records. However, if the "Use Genetics" check-box in the Population Parameters window is checked, then a Mate Accumulator selector is added to the Properties tab of the Reproduction event. If users then select the appropriate accumulator from this selector, then mate IDs can be logged by the Reproduction event. This will allow the Births report to capture the mate ID values.

The column headings provided for the Births Report are listed below:

- Replicate
- Time Step
- Event Name
- Population ID
- Group ID
- Parent ID
- Parent Trait Index
- Child ID
- Child Trait Index
- Mate ID

Deaths

The Deaths report produces a record for every death that occurred during a simulation. Deaths can result from Survival, Interaction, and Movement events.

The column headings provided for the Deaths Report are listed below:

- Replicate
- Time Step
- Event Name
- Population ID
- Group ID
- Deceased ID
- Deceased Trait Index

Genetics

The Genetics report produces a record for every birth event, assuming genetics as been implemented in a simulation. The child genome is fully enumerated, with a separate column being used for each locus. The order of the loci (from top to bottom in the Population Parameters Traits tab) will match the order in which the loci are written to the report (from left to right). That is, the top locus in the Traits tab will be the furthest left in the report. The bottom locus in the Traits tab will be the furthest right in the report, etc. For each locus in the child's genome, the alleles present at that locus are written in the form (x y).

The column headings provided for the Genetics Report are listed below:

- Replicate
- Time Step

- Population ID
- Child ID
- Mother ID
- Father ID
- Child Trait Index
- Child Genome (Enumerated)

Vitals

The Vitals report gathers information on the survival and reproductive rates observed during a simulation. Data is gathered exclusively from Survival and Reproduction events. The data in the report is sorted primarily by event type, and secondarily by time step. The Rate column gives the mean rate over the entire population. The Rates by Trait Index data has a separate column for each of the trait values used to stratify that event. The order of the trait combinations in the event (from top to bottom) will match the order in which the rates are written to the report (from left to right). That is, the top trait combination in the event will be the furthest left in the report. The bottom trait combination in the event will be the furthest right in the report, etc. The data in the Rate column will not equal the mean of the data in the Rates by Trait Index columns because the latter have already been averaged, and the number of individuals in each trait class is typically unequal. The number of columns in this report may vary by event, since events can employ distinct trait stratifications.

The column headings provided for the Vitals Report are listed below:

- Replicate
- Time Step
- Event Type (Reproduction or Survival)
- Event Name
- Population ID
- Rate
- Rates by Trait Index (Enumerated)

Movement

The Movement report summarizes a simulation's movement data. Multiple dispersals and explorations can occur in a Movement event, and the report quantifies how many were actually taken. Dispersal path lengths will typically exceed the straight-line distance separating the dispersal starting and ending points. But both are useful metrics. The report thus tracks both the path length, in hexagons (Hexagons Dispersed) and the euclidean distance separating the dispersal starting and ending points (Meters Displaced). The Movement report also records the ultimate outcome of the movement process. Outcomes include "start", "join", "floater", and "exploration". These correspond to starting a new group, joining an existing group, remaining a floater (the goal or goals

were not achieved), and constructing a new explored area. The "exploration" outcome will occur when the movement strategy is set to Construct Explored Areas.

The column headings provided for the Movement Report are listed below:

- Replicate
- Time Step
- Event Name
- Population ID
- Individual ID
- Individual Trait Index
- Number of Dispersals
- Hexagons Dispersed
- Meters Displaced
- Number of Explorations
- Hexagons Explored
- Outcome

The Trait Index values written out by the Movement report will all be zero when the Movement event does not include exploration (that is, when the Movement Strategy is set to Dispersal Only).

Ranges

The Ranges report principally tracks data on range size and quality. Range data are produced by Movement events (during exploration) and modified by Range Dynamics events. The distribution of range qualities (the Resources column) illustrate how successful individuals or groups have been at attaining the resources they require. The data on range size (Number of Hexagons), together with the quality and group size data can be used to evaluate whether the landscape is impeding or facilitating population growth, and the extent density dependence is influencing population dynamics. The actual range hexagons are listed at the end of each row.

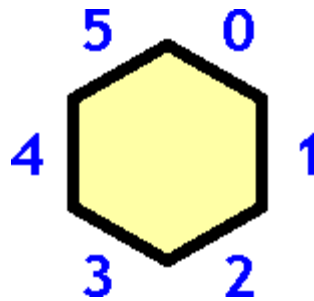
The column headings provided for the Ranges Report are listed below:

- Replicate
- Time Step
- Event Name
- Event Type
- Population ID
- Group ID
- Group Size
- Resources

- Number of Hexagons
- Hexagons (Enumerated)

Barriers

Barrier crossing can take place during both the dispersal and exploration components of Movement events. When a barrier is crossed, the result can be transmission (the individual crosses the barrier), deflection (the individual bounces off the barrier), or death. When individuals deflect off a barrier, they tend to skid. They do not behave like a billiard ball, which is why the term "deflection" is used instead of "reflection". HexSim barriers come in pairs, and are encountered when exiting a hexagon, not entering it (see the section on [Movement Event's Global Tab](#)). Individual interactions with barriers are associated with a specific hexagon (the hexagon being exited) and edge. The edges are numbered as shown in the figure below. The outcomes are also numbered. An outcome of 1 means the individual crossed the barrier (transmission). An outcome of 2 means the individual was bounced off the barrier (deflection). An outcome of 3 means the individual died when the encountered the barrier.



The column headings provided for the Barriers Report are listed below:

- Replicate
- Time Step
- Event Name
- Event Type
- Population ID
- Edge
- Hexagon
- Outcome

Stochasticity

Environmental stochasticity is simulated in HexSim by adding collections of rate tables to Survival, Reproduction, Transition, or Mutation events. When one or more collections are used, they must all have the same size. Then each time step, HexSim picks an index and uses it to draw a single rate table from the collection. The Stochasticity report

documents which indices were used as the simulation progressed. If N stochastic rate tables have been provided, they will be numbered 0 through N-1. An individual rate table is referred to as a Matrix, and an index value is referred to as a Matrix ID. Events that only have a single rate table will always have a Matrix ID of zero. The file created by the Stochasticity report is named using the key word "matrices", not "stochasticity".

The column headings provided for the Stochasticity Report are listed below:

- Replicate
- Time Step
- Event Name
- Event Type
- Population ID
- Matrix ID

Interactions

The Interactions report provides users with a full accounting of every trait change and death resulting from an Interaction event. The Location field will contain a hexagon drawn from the area in which the two individual's zones of influence overlap. If multiple populations exist in the simulation, a separate CSV report file will be generated for each population. The report for a given population will only contain interaction records for which it was the primary population.

The column headings provided for the Interactions Report are listed below:

- Replicate
- Time Step
- Event Name
- Location
- Primary Population ID
- Primary Individual ID
- Primary Trait Index (before)
- Primary Trait Index (after)
- Primary Status (alive or dead)
- Secondary Population ID
- Secondary Individual ID
- Secondary Trait Index (before)
- Secondary Trait Index (after)
- Secondary Status (alive or dead)

Dispersal Steps

The Dispersal Steps report tallies the number of times any individual dispersed from a hexagon to one of its immediate neighbors. Transitions for which the count is zero are not included in the report.

The column headings provided for the Dispersal Steps Report are listed below:

- From
- To
- Count

Dispersal Paths

The Dispersal Paths report tallies the number of time any individual dispersed between a pair of hexagons. The pair is made up of the starting point for dispersal, and the ending point for dispersal. If multiple dispersals are taken, the starting point is taken from the first dispersal, and the ending point is taken from the last dispersal.

The column headings provided for the Dispersal Successes Report are listed below:

- From
- To
- Successes
- Failures

The Successes column tracks the number of times that the disperser ended the Movement event as a group member. The Failures column tracks the number of times that the disperser ended the Movement event as a floater. If the Movement event being used employs the Dispersal Only strategy, then every dispersal path will necessarily be counted as a failure.

Projection Matrix

The Projection Matrix report constructs a population projection matrix from the log file IDV and BTH records. Users specify a population, a range of time steps, and select one or more traits. The report constructs an $N \times N$ matrix, where N is the total number of trait value combinations that can be made from the traits selected by the user. The matrix that is constructed is an array of probabilities that give the likelihood that individuals with one trait index (the columns) will transition into another trait index (the rows). In order to relate the trait indices to trait value combinations, users must create a [Census event](#), and in it they must select the exact same traits will be used to construct the projection matrix. The Census event will provide a mapping between the trait indices with the trait value permutations. Suppose, for example, that a projection matrix is built using just a stage class trait that has trait values 0, 1, and 2, corresponding to juvenile, subadult, and adult stages. The probability that appears in the bottom row, center column, would correspond to transitions from the subadult to adult stage classes.

Typically, the matrix that is constructed by this report is not a Leslie matrix, and it differs in significant ways from traditional population projection matrices. This difference stems from the temporal stratification of HexSim events. When a population vector is multiplied by a projection matrix, reproduction, transition, and mortality are simulated simultaneously. In HexSim, there can be multiple Survival, Reproduction, Transition, and other events that modify the population structure, and they operate at distinct points in time. The Projection Matrix report collapses time steps into transition records by comparing the before and after state of each individual. These transition records are tallied for a number of time steps indicated by the user, and the results are used to develop a matrix of transition probabilities.

The rows and columns of the projection matrices are assigned HexSim trait indices, from 0 to N-1. Each trait index corresponds to a trait combination. The View Combinations tool allows users to cross-walk between trait indices and the trait combinations they represent. The View Combinations tool can be accessed from the [Population Parameters Trait Tab](#), using the context menu associated with the Traits panel. This tool has also been integrated into Census events. It is necessary to use the View Combinations tool in order to interpret what trait values are represented by the rows and columns in the projection matrix.

In the most general case, a population projection matrix can be assigned the structure below (this example assumes a matrix dimension of four):

		FROM TRAIT			
		0	1	2	3
TO TRAIT	0	$(T_{0,0} + R_{0,0}) / X_0$	$(T_{1,0} + R_{1,0}) / X_1$	$(T_{2,0} + R_{2,0}) / X_2$	$(T_{3,0} + R_{3,0}) / X_3$
	1	$(T_{0,1} + R_{0,1}) / X_0$	$(T_{1,1} + R_{1,1}) / X_1$	$(T_{2,1} + R_{2,1}) / X_2$	$(T_{3,1} + R_{3,1}) / X_3$
	2	$(T_{0,2} + R_{0,2}) / X_0$	$(T_{1,2} + R_{1,2}) / X_1$	$(T_{2,2} + R_{2,2}) / X_2$	$(T_{3,2} + R_{3,2}) / X_3$
	3	$(T_{0,3} + R_{0,3}) / X_0$	$(T_{1,3} + R_{1,3}) / X_1$	$(T_{2,3} + R_{2,3}) / X_2$	$(T_{3,3} + R_{3,3}) / X_3$

For the T's and R's, the first subscript indicates the starting trait and the second indicates the ending trait. The T's represent numbers of transitions between trait combinations, and the R's represent numbers of newborns. The X's represent numbers of individuals having a specified trait.

The projection matrix data must be generated in part from pairs of log file IDV records. Let X_i refer to the number of individuals with trait combination i as recorded in the IDV record for time step S . Let Z_k refer to the number of individuals with trait combination k as recorded in the IDV record for time step $S+1$. A time step's starting population is represented by X , and its ending population is represented by Z . Individuals born during the time step will appear in Z but not X . Individuals who die during the time step will appear in X but not Z .

$T_{i,k}$ represents the number of individuals who had traits i and k respectively in two subsequent IDV records. The T data for time step S will not count individuals who are either born or die in time step S .

$R_{i,k}$ represents the number of newborns that ended a time step with trait k , whose parents started the time step with trait i . This R -data is gathered in part from log file BTH records. However, the trait values of parents and offspring at the time of birth are irrelevant. $R_{i,k}$ trait data must be computed by using the Z records to assign “to” traits (matching offspring ID’s) and using the X records to assign “from” traits (matching the parent ID’s). Newborns who die in the time step of their birth will not contribute to the computation. But parents who die (after reproducing) will still contribute to the computation of R .

If recruits themselves reproduce in their first year of life, then this second generation’s parents will not be stored in the IDV records. In this case, the parent trait data is obtained from the birth records. Time step zero data will not be used in the computations. HexSim stores a “terminal IDV record” at the end of the log file. This is used to compute the final set of transitions at the end of a simulation.

The projection matrix generated by this report will not typically resemble a Leslie Matrix or a traditional population projection matrix. This happens for multiple reasons, including because:

1. The “ i ” value in the $T_{i,k}$ and $R_{i,k}$ terms represents an individual’s trait index at the very beginning of a time step, *before any events have acted on the population*. This means these “ i ” values will be set to the trait index an individual had at the end of the previous time step (assuming it was alive then). For simple stage-based simulations (with no other traits), this will have the effect of shifting the columns in the projection matrix to the left.

For example, assume a simulation has 3 stage classes (stage-0, stage-1, stage-2), but no other traits. If individuals age once per time step, then the stage-1 reproductive rates will be experienced by individuals who were in stage-0 at the beginning of the time step. Therefore, these stage-1 individual’s reproductive output will be recorded in the matrix as stage-0 fecundity, and will appear as a non-zero entry in the upper-left cell of the matrix. If stage-0 individuals don’t reproduce, then observing a non-zero entry in the upper-left cell of the matrix might look incorrect. In fact, it is correct, and the same pattern will be observed for reproduction within the other stage classes.

This also complicates the recording of survival data. Suppose, for example that reproduction follows survival, which follows aging in the event sequence. An individual that is born in time step T will experience its first survival event in time step $T+1$. At the beginning of time step $T+1$, the individual will be in stage-0. Assuming it survives, it will end time step $T+1$ as a stage-1 individual. However, because the survival event followed aging, it will experience the stage-1

reproduction rate. This means that transitions from stage-0 to death will take place at a rate equal to the mortality for stage-1. And transitions from stage-0 to stage-1 will take place at a rate equal to the survival for stage-1. Thus the matrix element for transitions between stages 0 and 1 will hold the stage-1 survival rate. The matrix element for transitions between stages 1 and 2 will hold the stage-2 survival rate. Finally, the matrix element for transitions between stages 2 and 2 will also hold the stage-2 survival rate. The stage-0 survival rate is irrelevant in this case, because the survival event follows aging. This means that there will never be any stage-0 individuals present at the time when the survival event runs.

2. The reproduction rates in the projection matrix are likely to be lower than the rates in the reproduction event. This is because recruits must survive until the end of the time step to be counted. Also, the reproduction rates are computed by dividing the number of recruits, per trait index, by the number of individuals having that trait index at the start of the time step. But many of these individuals may not be reproductive, since they may be floaters, or too young, etc. Thus the actual rates supplied to a reproduction event will only be observed in the projection matrix when every individual reproduces and every recruit survives to the end of the time step.

Productivity

The Productivity report lists the total number of births, deaths, and births minus deaths, for the trait combinations specified by the user. This report is particularly useful when productivity is stratified using a location-based trait. In such cases, the births - deaths column will indicate the source-sink behavior of the individual locations. At steady-state, locations that have more deaths than births are functioning as demographic sinks. When a location at steady-state has more births than deaths, it is functioning as a demographic source.

Tallies

HexSim tallies are accessed through the Scenario drop-down menu. HexSim tallies are collections of count data recorded for each hexagon in a grid. These counts are incremented over all replicates, and all time steps specified by the user. HexSim tally data can be saved in comma-separated variable (CSV) format, or as HexMap (HXN) files. Both formats can be read by HexSim's [Display Spatial Data](#) tool, and therein displayed as a image, and saved to a bitmap, Shapefile, or CSV file.

Tally data can be placed into two functional groups:

Individual-Based Tallies

- Births
- Deaths
- Births Minus Deaths
- Interactions
- Occupancy
- Trait Counts

Hexagon-Based Tallies

- Barrier Deaths
- Barrier Deflections
- Barrier Transmissions
- Dispersal Flux

Hexagon-based tallies capture individual interactions with the landscape. When an individual encounters a barrier, it necessarily does so at a specific hexagon. The same is true for Dispersal Flux, which records the number of dispersers that pass through each hexagon. When these interactions take place, the target hexagon's tally gets incremented. Hexagon-based tallies are always attributed to a single hexagon, and they are always integer-valued.

Individual-based tallies record information specific to an individual. This information must then be attributed to one or more hexagons. There are two ways that individual-based tally data can be attributed to hexagons. Every individual can be assigned a single hexagon location. Tally data can then be written to these individual locations. But group members also have ranges, and tally data may also be assigned to ranges as a whole. When this is done, the each hexagon in a range of size N will be incremented by $1/N$. Individual-based tallies can thus be integer-valued, or real-valued.

The Interaction is an exception to the above rules. For individuals to interact, their zones of influence must overlap. But the zones of influence can be ranges, floater allocations, or explored areas. So there is no way to reliably distribute the tally over a range, even if the Tally Target is Groups Only. Thus for each interaction, a single hexagon is selected at random from the area in which the two zones of influence overlap, and this hexagon's tally is incremented.

The HexSim tally dialog allows users to select a starting and ending time step. Data will only be tallied for the time steps included within these bounds. This allows users to avoid recording information prior to the simulation's reaching steady state, or to tally data for just a small temporal window, etc. If multiple replicates were run, the tally is performed cumulatively across all replicates, but still respecting the starting and ending time steps.

Tally Targets

The Tally Target can be set to Floaters Only, Groups Only, or Floaters and Groups. The Births tally applies only to group members. The Dispersal Flux tally applies only to floaters. The other tallies can be applied to any of the three target groups.

If the Tally Target is Floaters or Floaters and Groups, then each individual will be assigned a single-hexagon location. The tallying process then involves incrementing these individual locations. For example, if a floater in hexagon 100 dies, then hexagon 100's death tally would be incremented by one. A floater's location is the last hexagon visited in the most recent dispersal. A group member's location is drawn from its range at random. When groups have multi-hexagon ranges, this makes the tallying process itself stochastic.

If the Tally Target is set to Groups only, then individual-based tally data will be distributed across ranges. When tally data is written to a group member's range, each range hexagon is incremented by an equal amount. For example, if a group with a range of 10 hexagon experiences a death, then each of the 10 hexagon's death tallies will be incremented by $1/10$.

The reason for averaging group member tallies across ranges is that the results more accurately reflect what actually took place during a simulation. Group members share their range resources, and movements or locations within a range are not explicitly simulated.

The reason for forcing tallies to increment a single hexagon when floaters are included is that to do otherwise produces misleading results. For example, if a floater death is assigned to a single hexagon, and a group member death is distributed across 10 hexagons, the floater death would appear much more significant.

Tally Types

Births

The Births tally records the number of births observed across the landscape. When the Tally Type is set to Births, the Tally Target is forced to Groups. This is because in HexSim, only group members may reproduce.

Deaths

The Deaths tally records the number of deaths observed across the landscape. Deaths are tallied when individuals die as a result of Survival or Interaction events. Deaths resulting from barriers are reported in the Barrier Deaths tally.

Births Minus Deaths

The Births Minus Deaths tally is equivalent to the Births tally minus the Deaths tally. The Births Minus Death tally captures the net productivity, or source-sink behavior of each hexagon. Hexagons with Birth Minus Death tallies less than zero are effectively demographic sinks. Those with tallies exceeding zero are effectively demographic sources. This is the only tally that can contain negative values.

Barrier Deaths

The Barrier Deaths tally illustrates where interactions with a barrier resulted in mortality.

Barrier Transmissions

The Barrier Transmission tally illustrates where barriers were successfully crossed.

Barrier Deflections

The Barrier Deflections tally illustrates where interactions with a barrier resulted in deflection.

Dispersal Flux

The Dispersal Flux tally is incremented each time an individual disperses into a hexagon. It characterizes the hexagon by hexagon dispersal frequency.

Interactions

The Interactions tally gets incremented each time an interaction takes place at a hexagon. Interaction locations are always a single hexagon picked at random from the area of overlap of the interacting individual's zones of influence.

Occupancy

For each individual corresponding to the Tally Target, the individual's location gets incremented by the Occupancy tally. If the Tally Target includes floaters, then the location is a single hexagon. For floaters, that hexagon is the endpoint of the last dispersal. For group members, the location is either the whole range, or a single hexagon drawn from the range at random. If the whole range is used, then each hexagon is incremented by $1/N$, where N represents the range size.

The Occupancy tally data is gathered in part from Census event's CIN log file records. This means that there must be at least one Census event for the Occupancy tally to produce a non-zero result. The location of the Census event (or events) in the life cycle will affect the tally results.

Trait Counts

For each individual corresponding to the Tally Target, if the individual has the indicated trait combination, then its location gets incremented by the Occupancy tally. If the Tally Target includes floaters, then the location is a single hexagon. For floaters, that hexagon is the endpoint of the last dispersal. For group members, the location is either the whole range, or a single hexagon drawn from the range at random. If the whole range is used, then each hexagon is incremented by $1/N$, where N represents the range size.

The Trait Counts tally data is gathered in part from Census event's CIN log file records. This means that there must be at least one Census event for the Trait Counts tally to produce a non-zero result. The location of the Census event (or events) in the life cycle will affect the tally results.

Display Spatial Data

The Display Spatial Data tool is accessed through the HexSim drop-down menu. This tool makes it possible to display tally data or generated HexMaps as map files. The Display Spatial Data tool has File and View menus. The items these menus contain are described below.

File Menu

- **Display HexMap**

This option calls up a file chooser. When a HexMap is selected, it will be displayed.

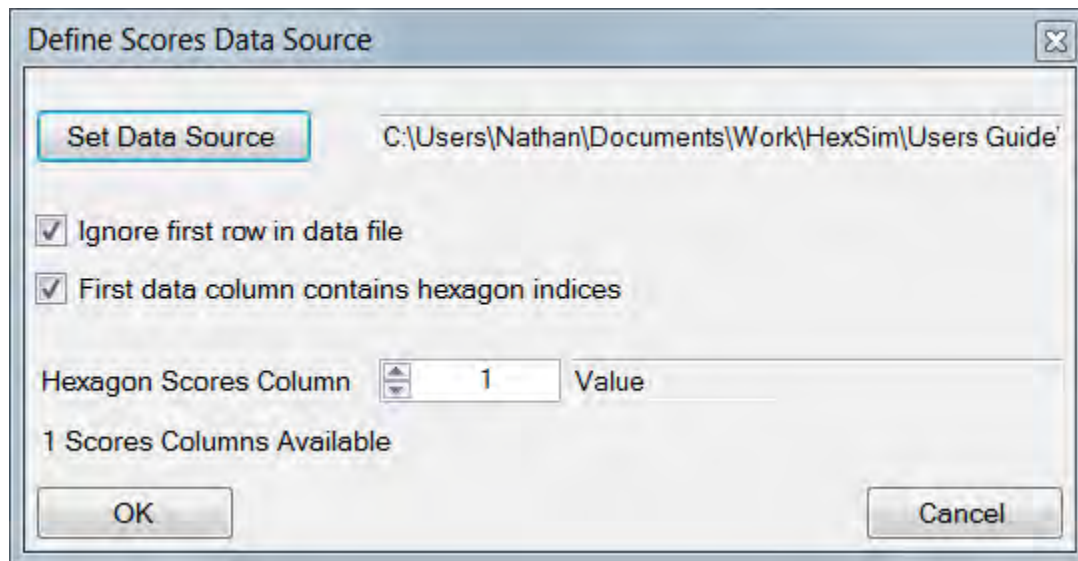
- **Display CSV File**

This option calls up the dialog window displayed below. A CSV file must be selected, using the Set Data Source button.

The first row in the CSV file may or may not contain a header. If the CSV file was written by HexSim, then the first row will contain a header. The check-box labeled *Ignore first row in data file* should be checked only if there is a header row in the CSV file.

The first column in the CSV file may or may not contain hexagon indices. If the CSV file was written by HexSim, then the first column will contain hexagon indices. If the file does not contain hexagon indices, then the row number is used as the hexagon index, with the first data row being assigned to hexagon 1. The check-box labeled *First data column contains hexagon indices* should be checked only if the first column does contain hexagon indices.

The CSV file being read may contain multiple data columns. Users are allowed to select which data column they would like to display, using the *Hexagon Scores Column* spin-dial. The selected column's header text (if it exists) is supplied to the right of the numeric field. The number of available data columns is displayed below the spin-dial.

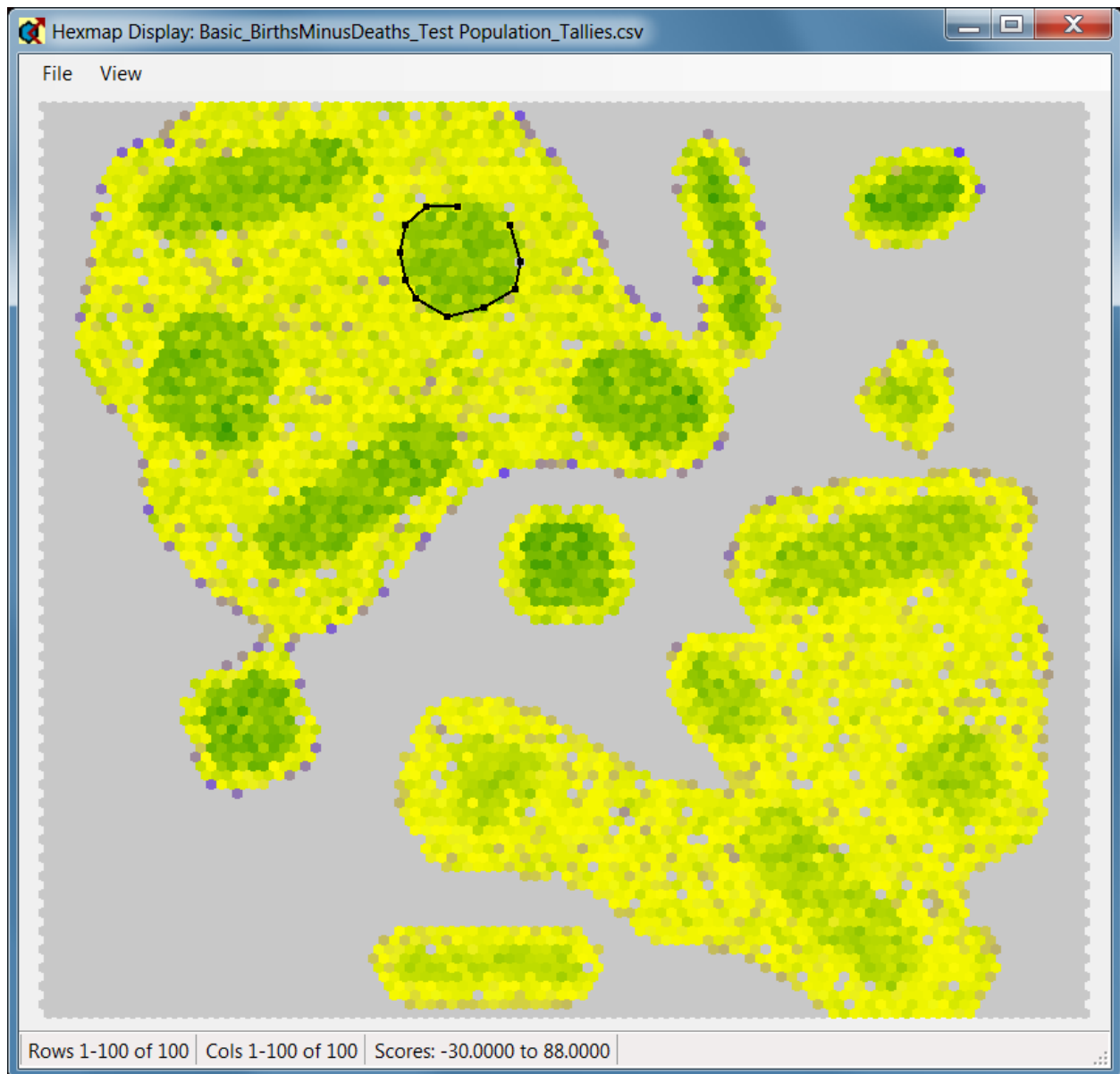


- **Save As**

Users may save the image as a bitmap, ESRI Shapefile, or CSV file. Saving an image as a CSV file is useful if it was originally in a HexMap format. Four bitmap resolutions are available (2, 4, 6, and 12 pixel hexagon widths). Higher resolution bitmaps will have a larger file size, but the hexagons within them will be better resolved.

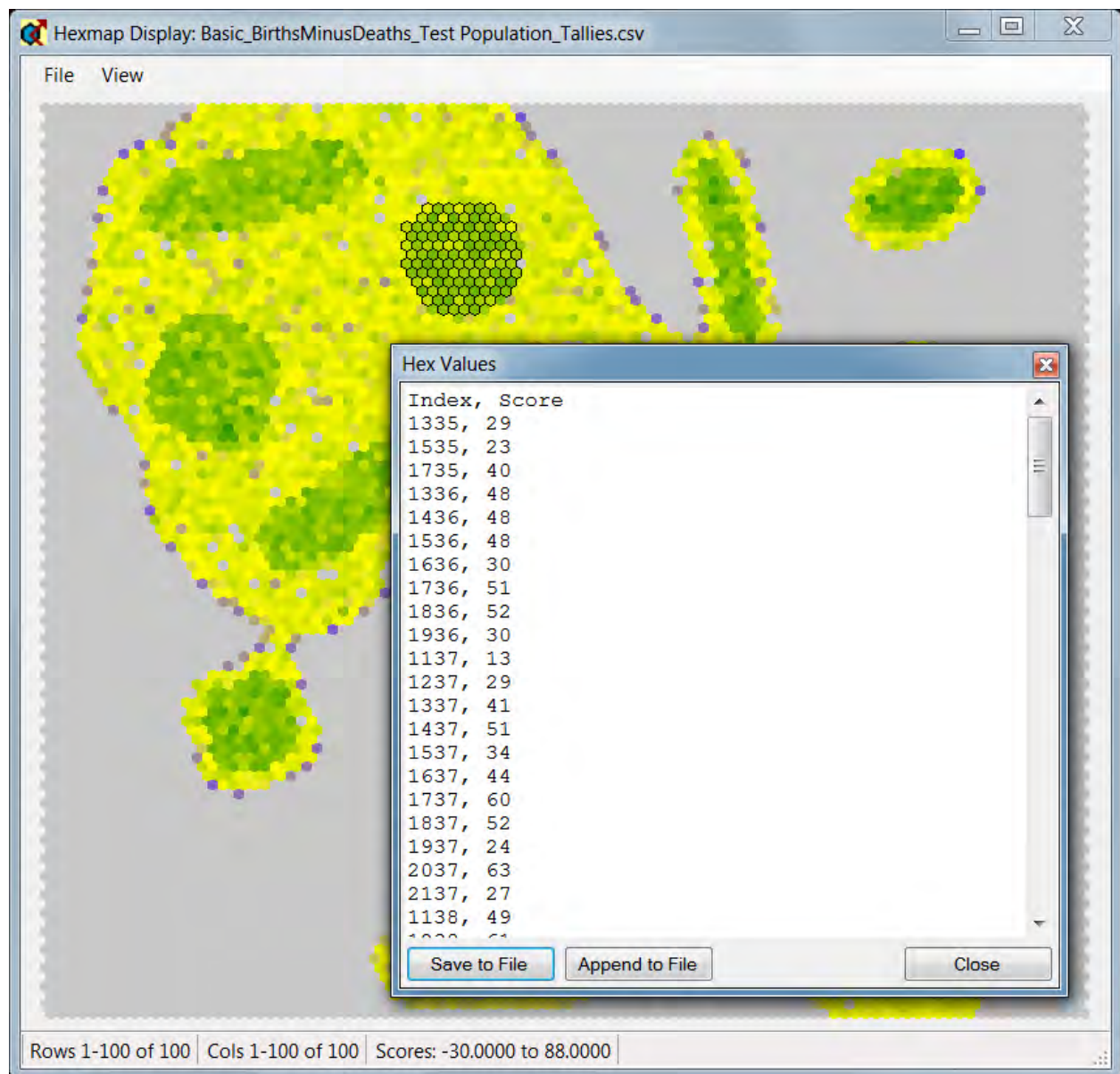
View Menu

The Display Spatial Data tool's View menu allows users to zoom in or out, to turn on or off the navigation window (right-clicking the image does this too), and to show or hide the grid lines. This menu also includes a Polygon Selection feature. After selecting this menu item, users click multiple times in the image so as to create a closed polygon (see image below).



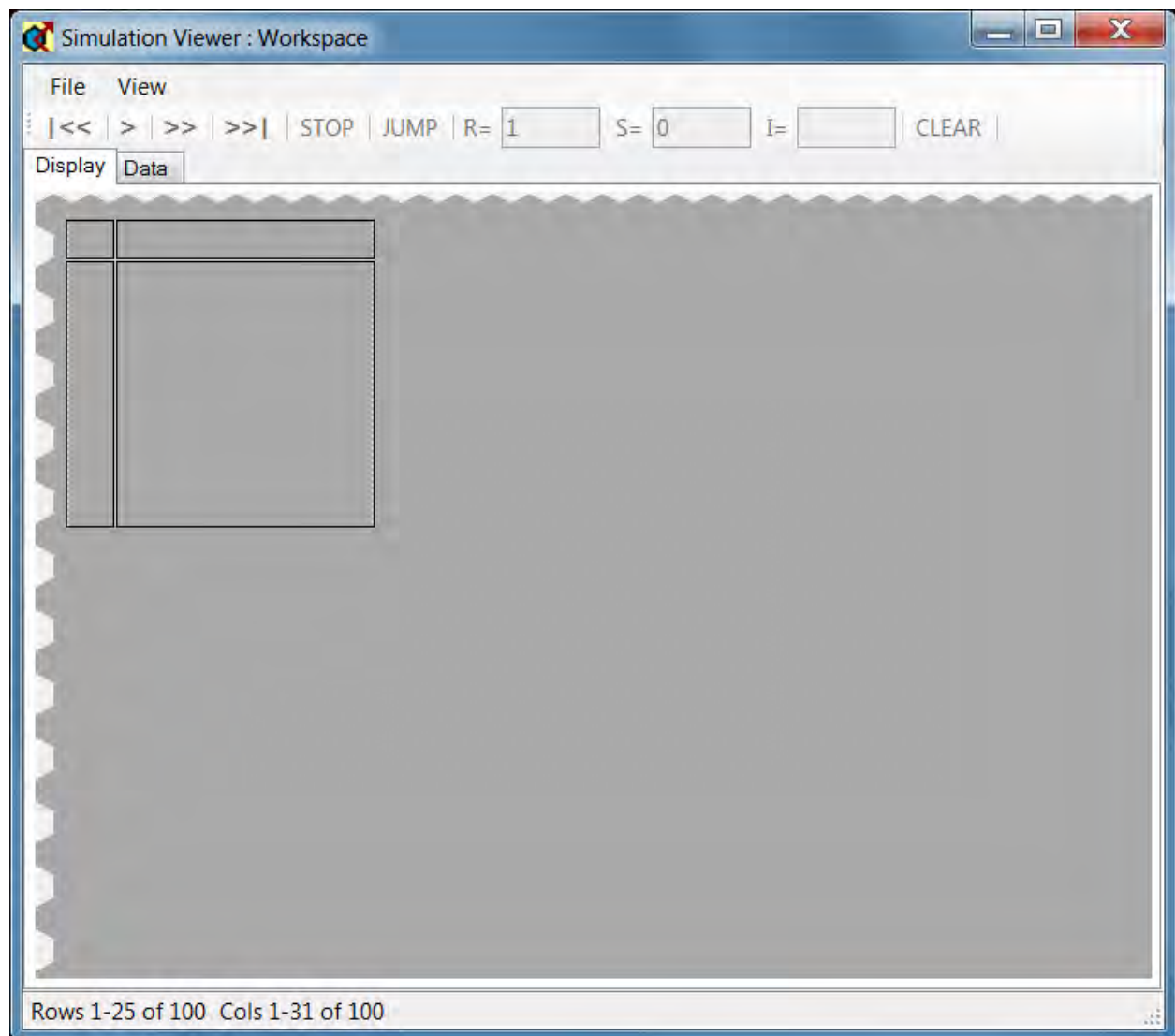
The backspace key can be used to remove polygon vertices. Once the polygon has been closed, two things happen (see image below). First, the polygon itself is replaced by with the outlines of each hexagon that it enclosed. Second, a window is displayed that shows the index and score of each selected hexagon. This data can subsequently be saved or appended to a file. Polygon Selection feature is useful for quantifying the value of areas of interest visible in a HexSim tally.

HexSim User's Guide



The Simulation Viewer

The HexSim Simulation Viewer is accessed through the HexSim menu. This tool allows users to visualize how groups or individuals moved around in the landscape as the simulation progressed. When the Simulation Viewer first comes up, it will appear similar to the image below.



The Data Tab

To begin, users must select the Data tab. The contents of the data tab are illustrated below:

Log File	Basic.log	
Population	Test Population	
Spatial Data	WKS: Matrix (gradient)	Restricted <input checked="" type="checkbox"/>
Barriers	>>> None Selected <<<	Restricted <input checked="" type="checkbox"/>
Visualization Type	Groups	

The simulation viewer uses a log file to create its imagery. Available log files are those present within the workspace's Results folder. Next, users must select a population from those used in the simulation that generated the log file.

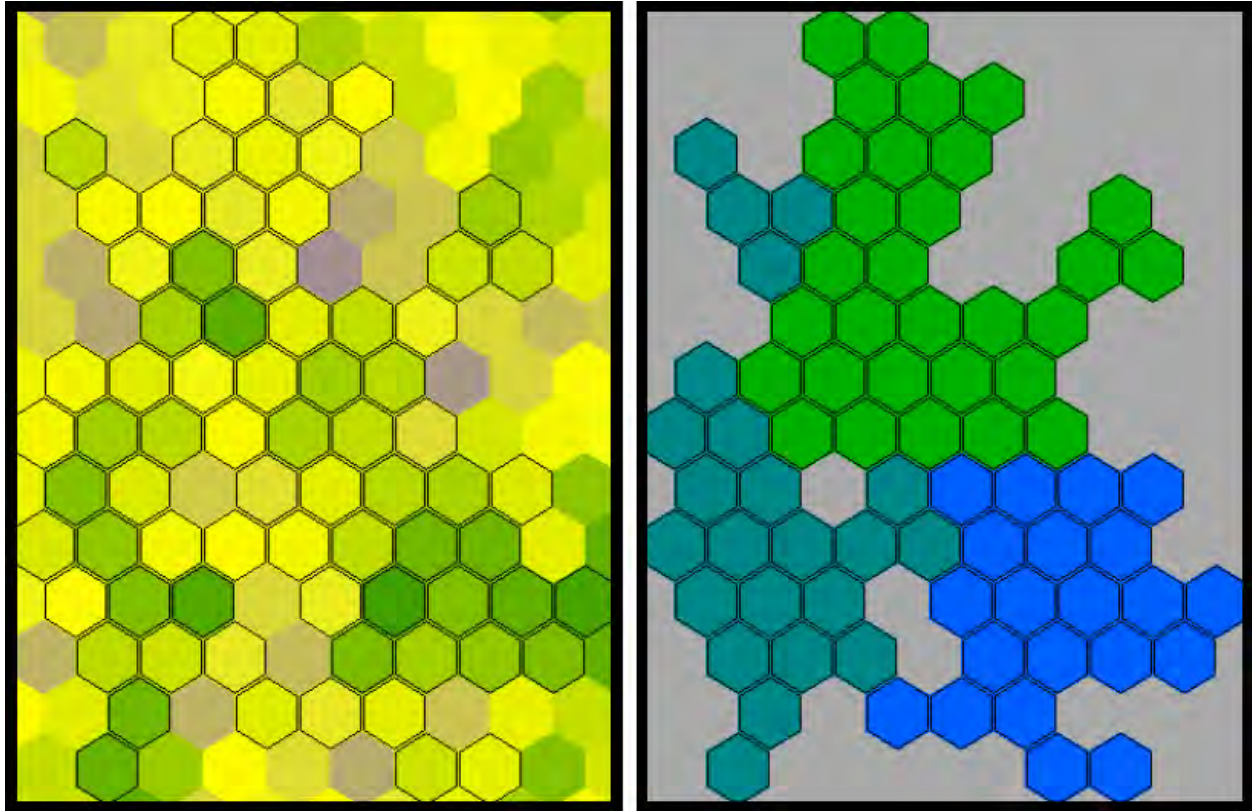
Spatial Data and Barriers may or may not be selected. If one or both are selected, they can be turned off later, using the Display tab's View menu. By default, the available Spatial Data and Barriers are restricted to those used in the simulation. If users uncheck the Restricted check-box (see figure above), then other HexMaps or Barrier maps present in the workspace can be selected. This allows results to be displayed on top of data that was not used in a simulation, or that was created during a simulation using a [Generated HexMap](#) event. Spatial Data and Barriers that are present in the workspace's Spatial Data folder are prefixed with WKS in the drop-down menus. Generated HexMaps are prefixed with GEN.

The Visualization Type menu allows users to choose between Groups, Movement, and Occupancy. These choices are discussed below.

Groups

When the Visualization Type is set to Groups, then the locations of groups will be displayed in the landscape. This process is handled differently depending on whether a HexMap is being displayed. A HexMap will be displayed if the Spatial Data menu is set to a HexMap, and if Show Hexagon Values is selected in the Display tab's View menu.

When a HexMap is being displayed, groups are indicated by drawing outlines around hexagons that are part of the groups' ranges. This is illustrated in the figure below, on the left. In this case, it is not possible to distinguish between adjacent groups. If a HexMap is not being displayed, then groups can be shown in color. In this case, HexSim uses four different colors to distinguish between adjacent groups. This is illustrated in the figure below, on the right.

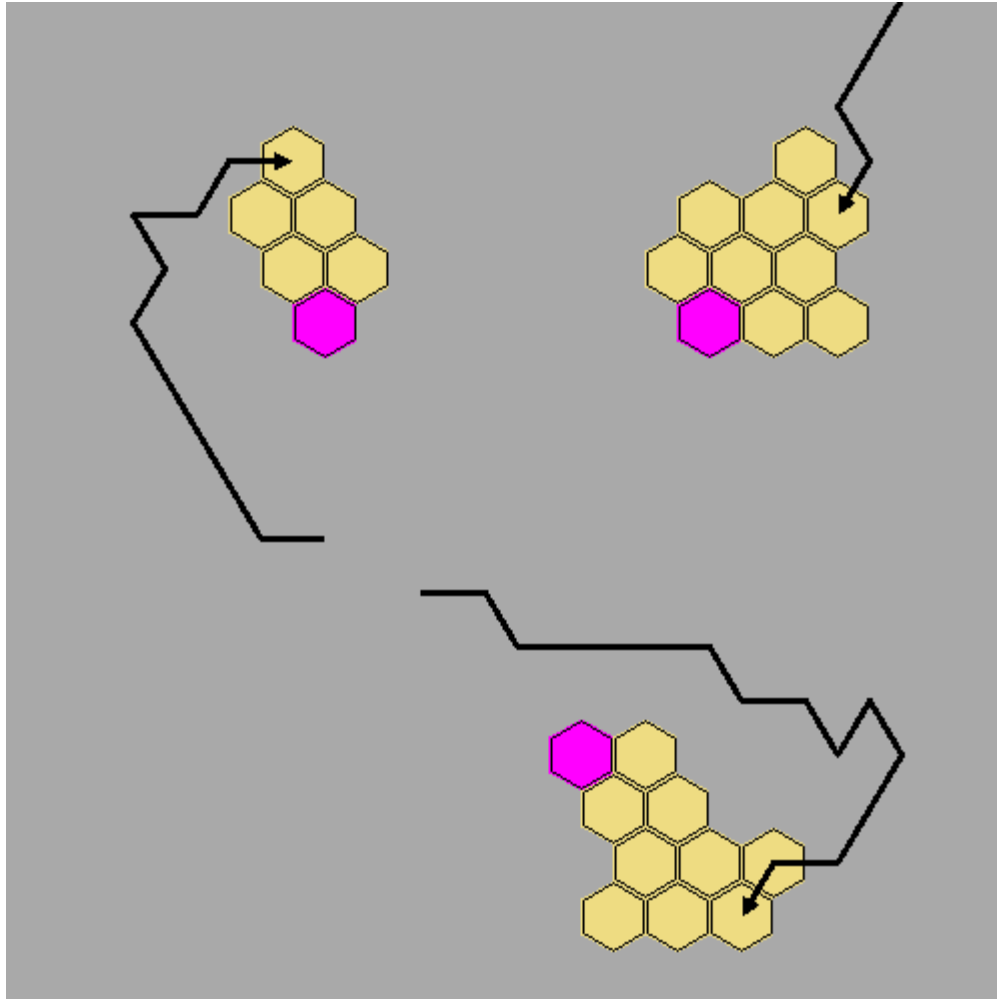


The Simulation Viewer display will update as the group locations change, creating an animated effect.

The data used to generate the group locations is that stored (in log files) by the exploration component of Movement events, and by Range Dynamics events. For every time step, the group display will be updated once for each Movement event that contains exploration, and for each Range Dynamics event. Thus the groups display may change multiple times per time step.

Movement

When the Visualization Type is set to Movement, users have a choice of displaying Dispersal, Exploration, or both. Exploration is depicted differently vary depending on whether a HexMap is being displayed. If a HexMap is visible, then explored areas are indicated using outlined hexagons, such as can be seen in the figure above, to the left. If a HexMap is not being displayed, then explored hexagons are shown in yellow, and ranges derived from them are shown in magenta (see figure below). In either case, dispersal paths are shown as black lines with an arrow head at the forward end.



The data used to generate the movement paths is that stored (in log files) by the dispersal component of Movement events. The explored areas and ranges are displayed using data stored by the exploration component of Movement events.

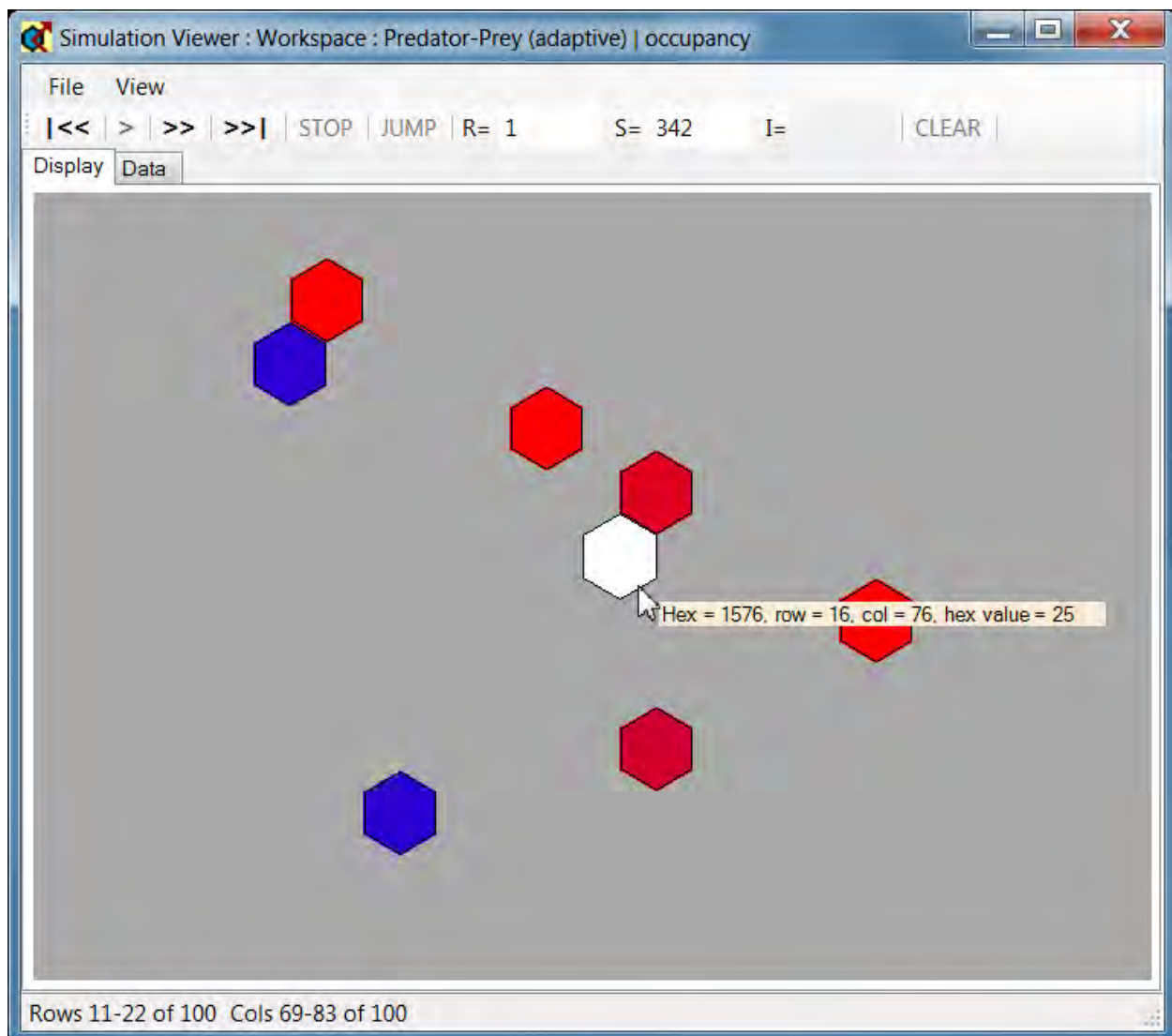
Occupancy

When the Visualization Type is set to Occupancy, users are allowed to select one or more traits, and trait combinations. A check-box is also supplied that displays the occupancy data for all trait combinations, hence for all individuals. If the All Trait Combinations check-box is not checked, then occupancy data will only be displayed for those individuals that have the selected trait combinations.

The data used to depict occupancy is stored (in log files) by Census events. These CIN records store the single-hexagon floater locations, and the full group members ranges. Floaters are shown at that stored location, but group members are *always* shown at a single hexagon picked at random from their range hexagons. This means that the occupancy animations generated by the Simulation Viewer are themselves stochastic. They can change each time it is run.

If multiple Census events are present in the life cycle, then occupancy data will be displayed for each one, within a single time step. However, logging can always be turned off for one or more Census events. If a Census event's logging is off, then that event will not add to the display of occupancy data in the Simulation Viewer.

When occupancy data is being displayed without a HexMap, color is used to illustrate the number of individuals occupying a hexagon. This color spectrum extends from red (one individual) to blue (the simulation-wide maximum), and cannot be changed. If a HexMap has not been supplied in the Data tab's Spatial Data selector, then selecting a hexagon will reveal how many individuals presently reside there. See the figure below for an illustration.



If a HexMap has been specified in the Data tab's Spatial Data selector, then the hexagon's score will be shown instead of the occupancy tally.

The Display Tab

The log file data describing group locations, movement, and occupancy is all shown in the Display tab. At the top of the Display tab are a series of navigation controls. These principal navigation controls include:

|<< Rewind. This button takes the viewer back to replicate 1, step 1. The keyboard shortcut **z** can be used to rewind.

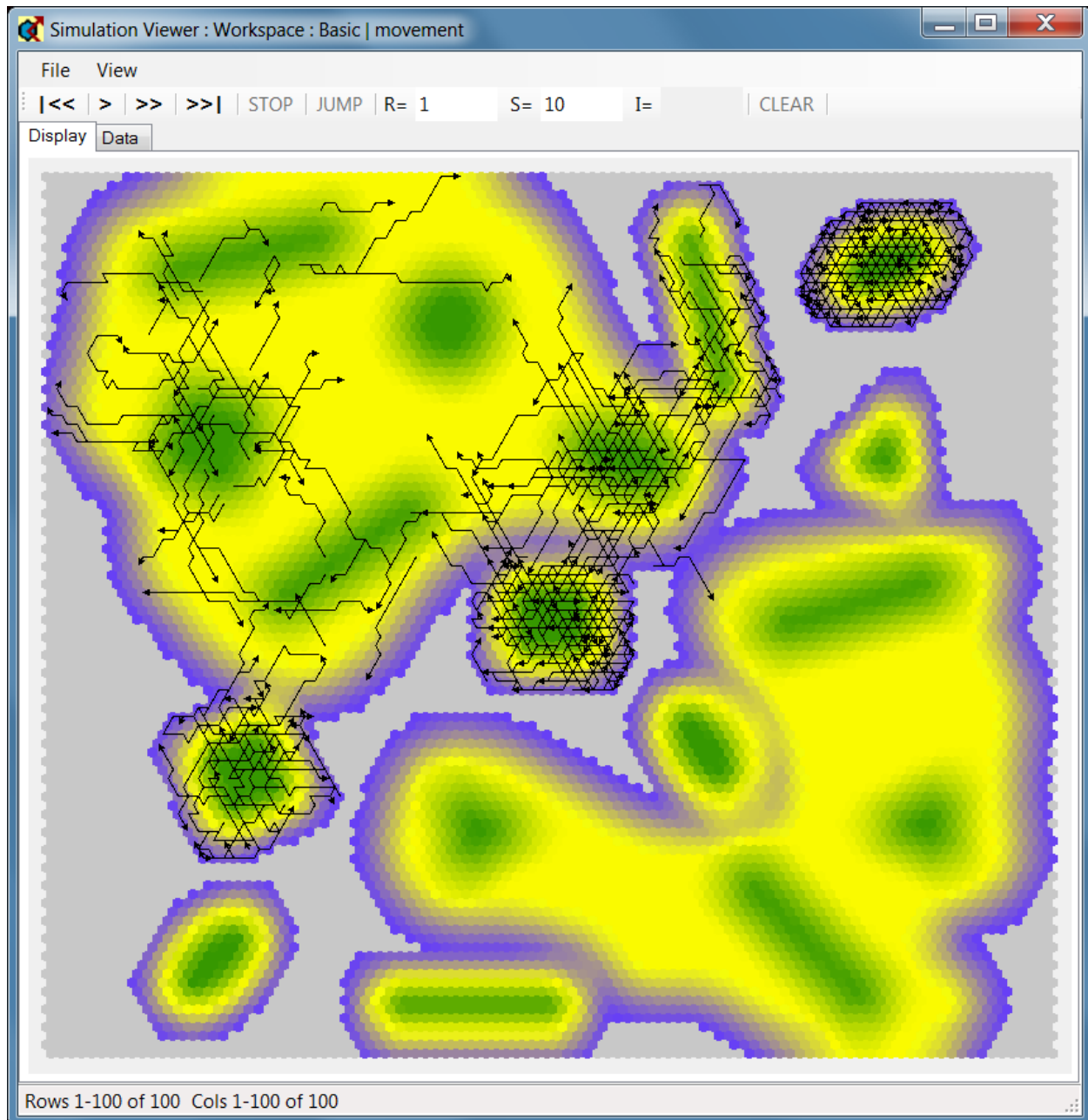
> Step by Individual. This button is only active when the Visualization Type is set to Movement. It will step to the next individual. The keyboard shortcut **i** can be used to step by individual. The keyboard shortcut **/** can be used to step by individual and simultaneously shift the rows and columns being displayed so that the individual can be seen. This is useful when only a fraction of the grid can be viewed at any one time.

>> Step by Event. This button steps to the next event that contains data of the type being displayed. The keyboard shortcut **t** can be used to step by event.

>>| Step to End. This button steps to the end of the simulation. The keyboard shortcut **e** can be used to step to end.

There are also **STOP**, **JUMP**, and **CLEAR** buttons, and counters showing the replicate (R), time step (S), and individual ID (I). The STOP button is only active after Step to End has been pushed. The CLEAR button is only active when Step by Individual has been pushed. The JUMP button will move the simulation to a replicate and time step specified by the user. First, change the values in the R or S fields, then push the JUMP button.

As the simulation progresses, the R and S counters will be updated. These fields indicate what replicate and time step is currently being processed. If multiple copies of the event being displayed exist in the life cycle, then there may appear to be several distinct actions taking place within a time step. When the Step by Individual button is being used, the I counter will display the individual ID. An image of the Simulation Viewer showing dispersal paths is presented below.



If users click on a hexagon and hold down the mouse, the hexagon ID, row, column, and score will all be displayed. In addition, the color ramp being used will be shown at the bottom of the screen, and a bar will be placed where this hexagon's score falls in the spectrum. An exception exists for the case in which the Visualization Type has been set to Occupancy, and no HexMap has been specified. Then the color scale will be used to display occupancy rates. And when a hexagon is clicked and the mouse held down, the occupancy rate will be displayed in place of the score.

The File Menu

The File menu contains just an exit option. Other options that might have been located here have all been placed into the Data tab.

The View Menu

The View menu allows users to:

- Change the color scheme used to display HexMaps
- Zoom In and Out
- Toggle the Navigation Window on and off
- Toggle the HexMap display on and off
- Toggle the Barrier Map display on and off
- To the extent possible, shift the rows and columns so that a specific hexagon appears in the center of the display.

Utilities

Utilities

[Workspace Construction](#)

[HexMap Generation](#)

[Taking Snapshots](#)

[Using Recover](#)

[Global Assignment](#)

[Show Usage](#)

[Importing Data](#)

[Batch Processing](#)

[Sensitivity Analysis](#)

[Command Line Options](#)

Workspace Construction

The Create Workspace tool is accessed from the HexSim drop-down menu. This tool is used to construct a new workspace and grid. The workspace itself is just a collection of folders (see HexSim Workspaces in the section titled [HexSim Fundamentals](#)). The grid file at the top of the workspace defines the workspace dimensions and hexagon size. Only one grid file, and hence one set of grid parameters can exist per workspace. An image of the Create Workspace tool is shown below.

Create Workspace

New Workspace Directory Browse

C:\Users\Nathan\Desktop

New Workspace Name

Demonstration

Hexagon Area (hectares) 0.86602540

Hexagon Width (meters) 100.0000

Grid Configuration

Number of Rows 100

Hexagons per Row 100


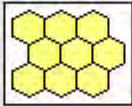
Total Hexagon Count 10000

Grid Dimensions (meters)

Height 8689.1215

Width 10050.0000

☐ Narrow ☒ Wide

OK Cancel

Users select a directory and workspace name. This specifies where the results will be written. Next, either a hexagon area (in hectares) or width (in meters) must be provided. Changes to one of these values will automatically be propagated to the other. Finally the grid configuration just be established, including selection of a Narrow versus Wide

layout. The total number of hexagons, plus the grid height and width are displayed in read-only fields.

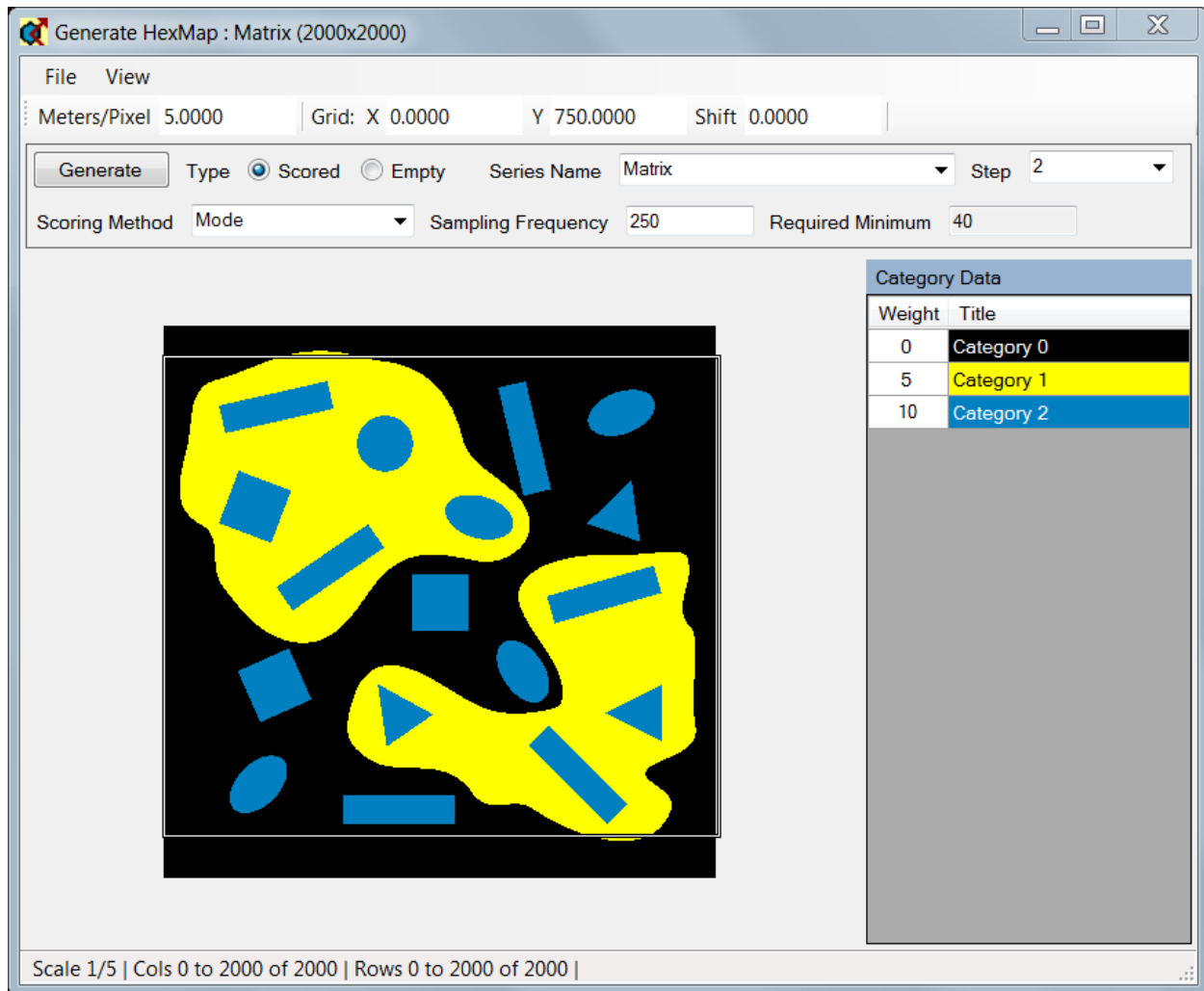
Workspace generation is essentially instantaneous, and the initial empty workspace takes up just a tiny amount of space. So experimenting with different grid configuration parameters is simple and fast. The hexagon size is typically set so that the smallest items of concern in the landscape can be resolved. If the hexagon size is too large, each hexagon can end up containing a vast range of habitat types. If the hexagon size is too small, then tallies become unwieldy, and less and less of any HexMap can be displayed on the screen at once. Several research projects have successfully used grids exceeding 7 million hexagons. But these extreme cases can be cumbersome.

The conversion from hexagon width W (in meters) to hexagon area A (in hectares) can be performed by hand from the relationship: $A = W^2 \times \sqrt{3} \div 20,000$.

HexMap Generation

There are several ways to generate HexMaps. Users can import ESRI Shapefiles or CSV files. Users can also use the Generate HexMap tool to construct both empty and attributed HexMaps. Empty HexMaps can be attributed later using the Hexagon Editing tool in the HexMap viewer. Using the Generate HexMap tool to construct an attributed HexMap means supplying a bitmap and specifying several different parameters. An image of the Generate HexMap tool is shown below. The menus and parameters in the Generate HexMap tool are discussed in the sections that follow. The Generate HexMap tool is accessed from the HexSim menu.

The generation of attributed HexMaps works by orienting the HexSim grid (a workspace's array of hexagonal cells) on top of a bitmap. Assignments are then made between each hexagon and the collection of bitmap pixels its coincident with. The bitmap pixels that are circumscribed by a hexagon are sampled, and from this sample a score is generated. That score is assigned to the hexagon, and the process continues until the entire grid has been attributed. The HexSim grid is represented by its bounding box. This is the white box appearing in the figure below. The grid may extend beyond the bitmap. But those areas outside the bitmap will be treated as if they had been assigned category weights of zero.



The File Menu

Generate HexMap's File menu has two items. They are described below:

- **Set Bitmap**

This menu item is used to select a bitmap. The bitmap will be sampled, and the samples will be used to attribute the hexagon scores. Only 8-bit uncompressed bitmaps with an embedded colormap may be used. These bitmaps will have up to 256 different categories. XnView (www.xnview.com) is a free raster viewing and editing tool that can convert just about any raster image into a format that HexSim can use.

- **Get Parameters from HexMap**

This menu item is used to retrieve the values of the Meters / Pixel and Grid X and Y parameters from a previously generated HexMap.

The View Menu

The View menu provides access to a number of useful routines. They are all described below:

- **Zoom In / Zoom Out**

These menu items increase or decrease the screen resolution of the bitmap image. The + and - keys can also be used to zoom in and out.

- **Pan by Half Screens / Pan by Whole Screens**

If a portion of the bitmap fills the entire screen, then users may pan through this image using the arrow keys. This menu item determines whether panning is by whole- or half-screens.

- **Set View to Grid Corner**

The HexSim grid is displayed as a white box on top of the bitmap. If the grid is not visible, this menu item will automatically pan as necessary to bring the HexSim grid's upper-left corner into view. The <Alt> G keyboard shortcut will accomplish this as well.

- **Build Histogram**

This menu item allows users to build a histogram of pixel frequencies for the entire bitmap, or for just the portion of the bitmap enclosed within the grid's bounding box.

- **Show / Hide Generate Controls**

This option simply shows or hides the generate controls. This can be useful when working with a small display. The <Alt> B keyboard shortcut will accomplish this as well.

- **Show / Hide Categories**

The Category Data panel is used to assign weights to the bitmap categories. This menu item opens and closes the Category Data panel. The categories are drawn from the bitmap's colormap. HexSim works with 8-bit bitmaps, and these always have 256-element colormaps. However colormap entries that are not assigned to bitmap pixels are, by default, not displayed in the Category Data panel. The <Alt> C keyboard shortcut will accomplish this as well.

- **Display Category Indices**

This menu item is only available when the Category Data panel is visible. When selected, it adds a column of category indices to the Category Data panel.

- **Display Inactive Categories**

This menu item is only available when the Category Data panel is visible. Normally, HexSim only displays categories that are assigned to at least one bitmap pixel. Unused categories are not displayed in the Category Data panel. When selected, this menu item forces all 256 categories present in the bitmap to be displayed in the Category Data panel. Unused categories are assigned a weight of -1.

Bitmap Attributes

The Generate HexMap tool can only read an 8-bit uncompressed bitmap having an embedded colormap. Such bitmaps can store up to 256 different landcover classes. Bitmaps may be as large as necessary. The only limitation is the computer's memory. Many free tools exist that can convert arbitrary raster formats into 8-bit bitmaps. An outstanding example is XnView (www.xnview.com).

The Bitmap Parameters

- **Meters / Pixel**

The HexSim grid was defined when the workspace was constructed, and it has a fixed extent in meters. The extent of the grid is shown as a white box (see figure at the top of this page). The Meters / Pixel parameter defines the extent of the bitmap. As bitmap pixels increase in size, the bitmap's extent grows relative to that of the grid. This has the effect of making the grid appear to shrink on the screen. When real landcover data is being used, the number of meters per pixel should be known. For example, this value is typically 30 for Landsat Thematic Mapper (TM) data.

Hexagons may be large or small relative to the size of a bitmap pixel. There is no limit on the number of pixels allowed per hexagon, or the number of hexagons allowed per pixel.

- **Grid X and Y**

The origin of HexSim's grid is the lower-left corner of its bounding box. Thus the lower-left corner of the grid's bounding box is always set to (0, 0) meters. The Grid X and Y parameters control how far the grid origin is offset from the bitmap's lower-left corner. When the Grid X and Y are set to (0, 0) meters, the grid bounding box's lower-left corner will be aligned with the bitmap's lower-left corner. As the Grid X and Y parameters are increased, the grid origin will move towards the bitmap's upper-right corner. These parameters simply allow users to change the placement of the grid relative to the bitmap.

The Grid X and Y values can also be changed by right-clicking in the bitmap. When this is done, the upper-left corner of the grid's bounding box is moved to the mouse cursor. Finally, the Grid X and Y values can be changed using the keyboard arrow keys. See the section below titled Shifting.

- **Panning**

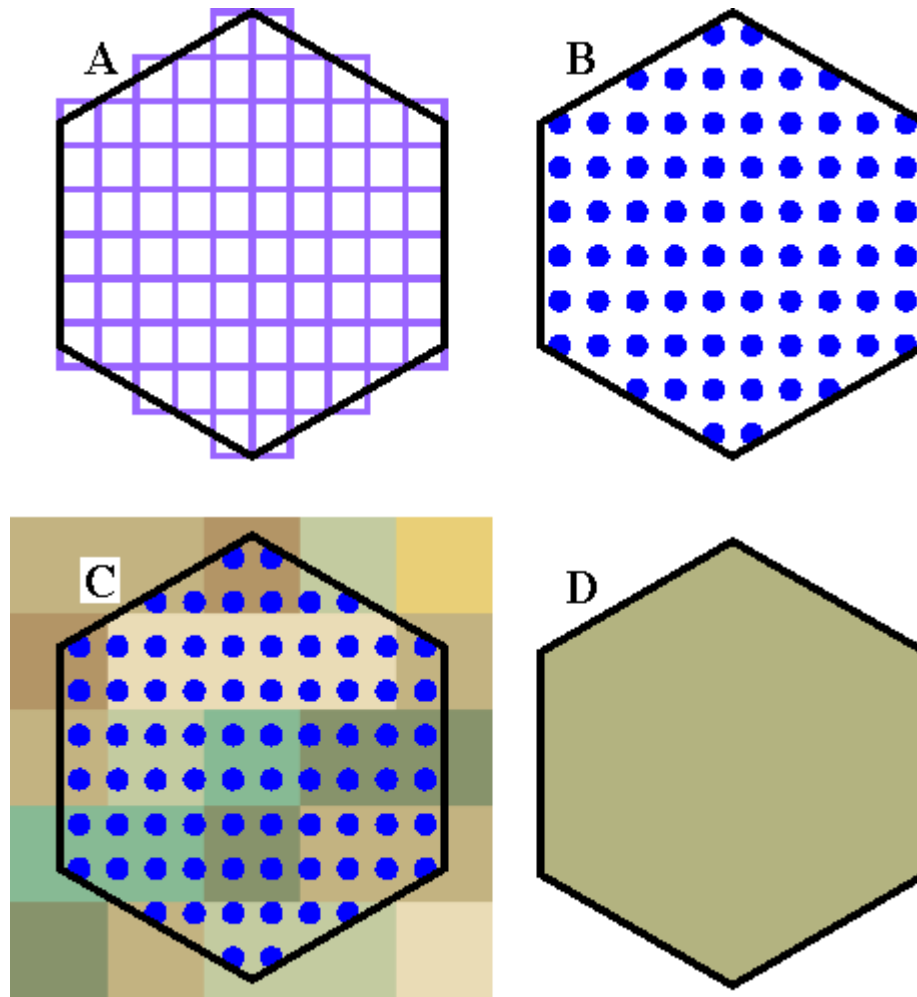
When only a portion of the bitmap can be seen at once, users can pan right and left, or up and down through the image. Panning is done using the arrow keys. A View menu item allows users to select whether panning will increment by full or half screens.

- **Shifting**

Shifting is the process of using the arrow keys to move the grid relative to the bitmap. Users use the Shift parameter to set the amount (in meters) that the grid will be shifted. Then the arrow keys are used to do the shifting, while the <Shift> key is held down. If the <Shift> key is not held down, then the arrow keys will pan across the bitmap image.

The Sampling Parameters

The Generate HexMap tool assigns scores to hexagons by sampling a bitmap image. Users provide weights for each bitmap category (e.g. landcover class), align the bitmap and grid properly, and supply a scoring method and resolution. The sampling process breaks each hexagon into an array of rectangular cells, and then samples the bitmap once per cell. The samples are collected and used to assign the hexagon score. This process is repeated for each hexagon in the grid. The figure below illustrates the steps involved in sampling a bitmap.



Initially, each hexagon is broken up into an array of rectangles. These rectangles are the sampling units. The sampling units are rectangles because their height and width are set such that an integer number fit along each hexagon's width and height. The Sample Frequency parameter is equal to the actual number of these rectangles that fit across each hexagon (width and height). Figure A above illustrate this step, and correspond to a Sampling Frequency of 10. The default Sampling Frequency is 100.

Next, the center of each sampling unit is identified. These are the sampling points. Once the sampling points have been identified, the sampling units themselves are no longer needed. Figure B above illustrate this step.

Finally, the sampling points are used to sample the bitmap (figure C, above). Unlike the figure, the actual sampling points have no extent, and each falls over at most one bitmap pixel. That pixel's weight (from the Category Data panel) is assigned to the sampling point. If the sampling point falls outside the bitmap, it is assigned a value of zero. Finally, the algorithm selected in the Scoring Method menu is used to develop a single hexagon score from the array of sampling point values (figure D, above).

The Required Minimum parameter is always set equal to the sampling frequency that guarantees every bitmap pixel gets sampled at least two times. The Required Minimum will change with the bitmap Meters / Pixel parameter. The Generate HexMap tool cannot be used to construct an attributed HexMap unless the Sampling Frequency is set equal to or greater than the Required Minimum.

The Scoring Method can be set to Mean, Mode, Sum, or Occurrence. These methods are used to compute a hexagon score from the sampling point values as described below. A sampling point's value is always set equal to the category weight assigned to the pixel below it.

- **Mean**

The value of each sampling point is summed. The sum is divided by the number of sampling points. The resulting value is assigned to the hexagon.

- **Mode**

The hexagon is assigned a score equal to the value most frequently observed among the sampling points. Ties are settled randomly.

- **Sum**

The value of each sampling point is summed. After this has been done for every hexagon, the results are rescaled so that they range between the lowest and highest category weights supplied by the user. This rescaling is necessary because hexagon scores have no units. The Mean and Mode operators necessarily produce results falling between the minimum and maximum category weights, so this choice for scaling ensures that the Mode produces comparable results. The rescaling is done after the sampling has finished for the entire map, so it preserves the relative magnitude of each hexagon's score.

- **Occurrence**

If any of the sampling points have a value greater than zero, then the hexagon is scored 1.0. Otherwise the hexagon is scored 0.0.

The Category Data

Before an attributed HexMap can be constructed, category weights must be supplied. The bitmap categories can be thought of as the subset of the bitmap colormap entries that have been assigned to at least one pixel in the bitmap image data. The category weights are relative importance values. When sampling points (described above) fall over a pixel, they assume the category weight assigned to that pixel.

Bitmap categories will typically reflect landcover classes. But they can represent any number of other spatially distributed landscape attributes, such as the extent of toxins or

competitors. The category weights are used to indicate how good or bad each category is relative to the others.

Categories are assigned the colors present in the bitmap's colormap. Users may also add category names. The default weights are zero, and the default names are Category N, where N is the category index. The category weights and names will be stored in any HexMap built from the bitmap.

The Category Data panel can be displayed or removed using the Show / Hide Categories View menu item. There are also View menu items that turn on and off the display of category indices and inactive categories (see figure below). The category indices are just a read-only column of index values. Displaying the category indices can be useful if the category titles have been customized. Inactive categories are those colormap entries that have not been assigned to any pixels in the bitmap image data. Weights assigned to inactive categories would have no effect on the generation of hexagon scores, so HexSim makes the Weight field inactive for inactive categories.

Category Data		
	Weight	Title
0	0	Category 0
1	0	Category 1
2	0	Category 2
3	-1	
4	-1	
5	-1	
6	-1	
7	-1	
8	-1	
9	-1	
10	-1	
11	-1	
12	-1	
13	-1	
14	-1	
15	-1	
16	-1	
17	-1	
18	-1	

Zero All Weight Values
Get Weights From Series/Step ...
Get Titles From Series/Step ...
Ramp Selected Weights
Ramp Weights By Index

The Category Data panel has a context menu (see figure above). This context menu lets users perform the following operations:

- **Zero All Weight Values**

This context menu item will set every category weight to zero.

- **Get Weights From Series / Step**

Users select a time step (a single HexMap) from a spatial data time series, and the category weights used to construct that time step will be loaded into the data table. If mismatches are found, a warning message will come up. HexMaps imported from CSV or Shapefile data, HexMaps that were not originally attributed (e.g. built as empty and then edited), and those constructed by the Generate HexMap event will not contain any category weight data.

- **Get Titles From Series / Step**

Users select a time step (a single HexMap) from a spatial data time series, and any category titles stored in that HexMap will be loaded into the data table. If mismatches are found, a warning message will come up. HexMaps imported from CSV or Shapefile data, HexMaps that were not originally attributed (e.g. built as empty and then edited), and those constructed by the Generate HexMap event will not contain any category title data.

- **Ramp Selected Weights**

Users must first multi-select some set of categories. Then this menu item will call up a dialog window that allows users to specify an initial value, and an increment. The selected weights will be assigned weights based on these parameters. This feature cannot be used if inactive categories are being displayed.

- **Ramp Weights By Index**

This menu item will assign each category a weight equal to its index value. This feature cannot be used if inactive categories are being displayed.

Building a HexMap

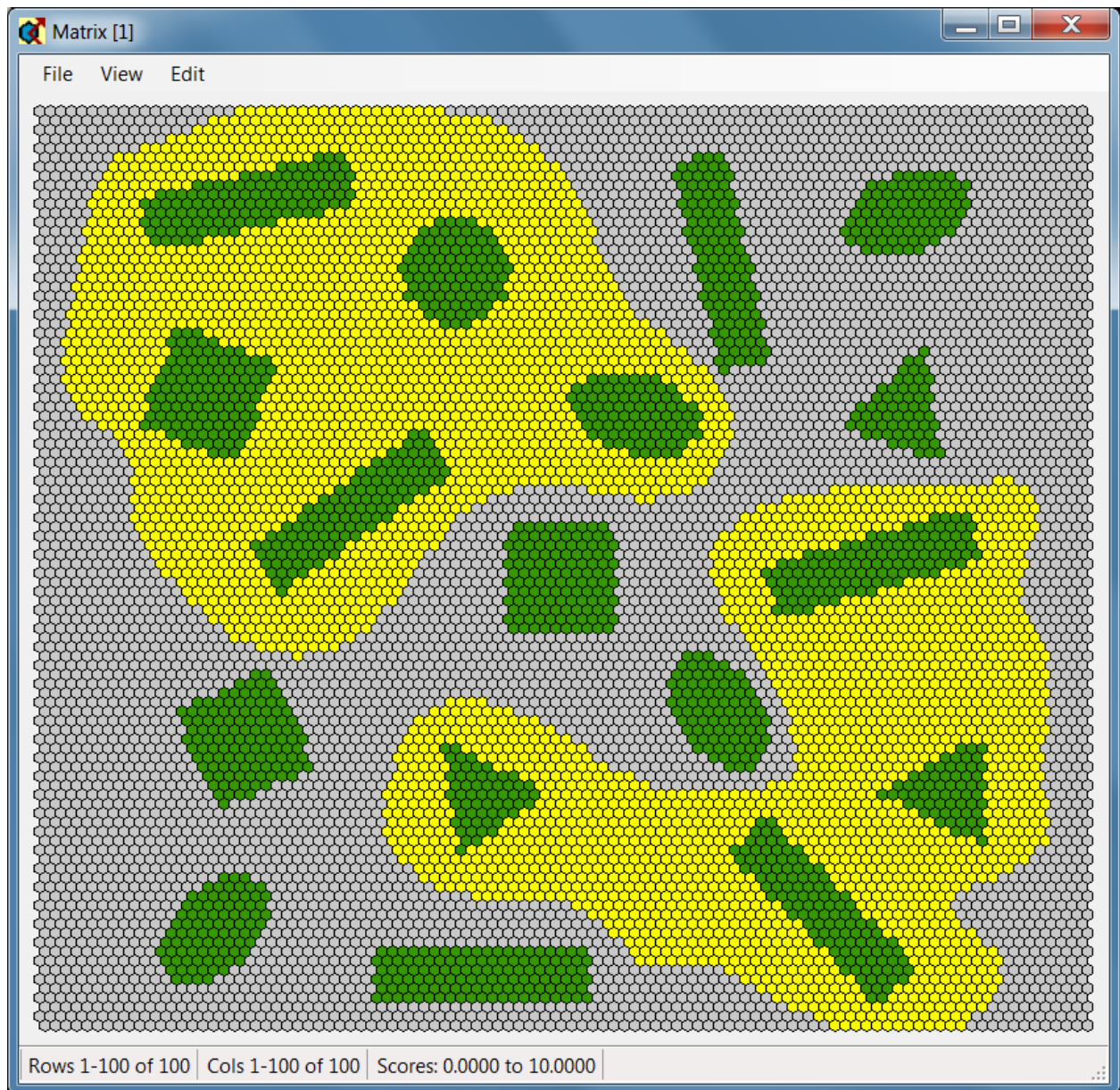
Once the grid has been oriented relative to the bitmap, and the bitmap parameters, scoring method, sampling frequency, and category data have been supplied, users can generate a new HexMap. A Series Name and Step must be supplied. The Series Name may be new or existing. If it is new, then the Step will be forced to 1. If the Series Name has already been used, then the Step must be set to a value not already present in the workspace spatial data.

To build the new HexMap, users must then push the Generate button. A progress bar will indicate how much of the process has completed. (The progress bar sometimes comes up behind other windows, and must be brought to the foreground manually.)

Users should feel free to supply large Sampling Frequencies. The HexMap generation process can always be aborted if the progress bar is moving too slowly.

Once the process of building a HexMap has started, there are two steps necessary to terminate it. First, users must dismiss the Generate HexMap window. This step is required because the main HexSim interface cannot be accessed while the Generate HexMap window is up. The second step involves clicking on the HexSim drop-down menu, and selecting Stop Dependent Processes. This will kill the HexMap Generation process (and any other child process spawned by HexSim.exe).

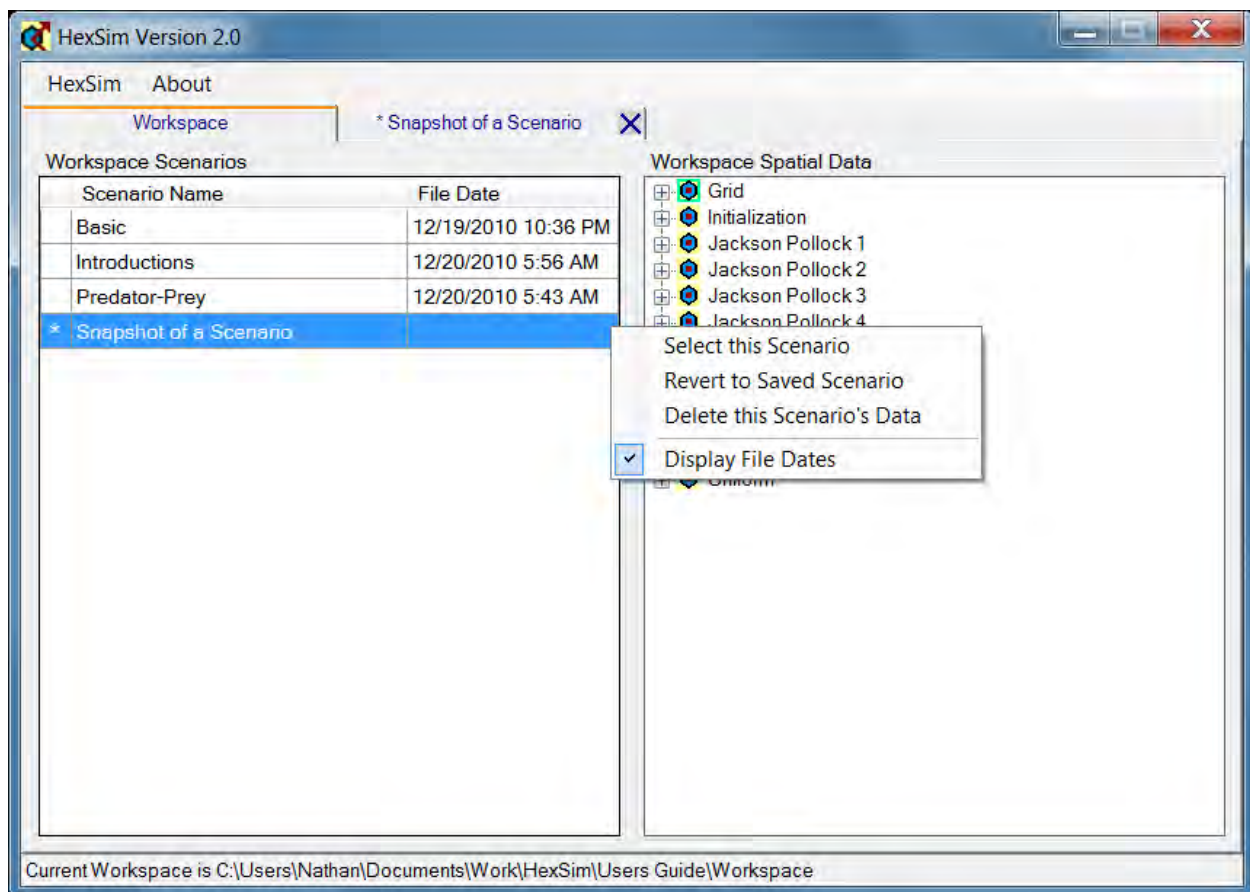
Multiple HexMaps can be built sequentially. Once the construction process has been completed, then the Generate HexMap window should be dismissed. Then users may examine the new HexMaps by selecting them from the Workspace Spatial Data panel, in the HexSim Workspace tab. A simple HexMap appears in the illustration below.



Taking Snapshots

HexSim's engine can be launched from the command line, with the scenario file passed as an argument. The model engine does not validate scenarios, and thus users may run simulations using flawed scenarios. Doing so could crash the program, or worse, produce results with errors, some of which might be subtle. To keep this from happening, the HexSim GUI will not save scenarios to the disk until they have passed its extensive error checking. This scheme necessitates an alternative to saving scenarios, so that works-in-progress don't have to be completed before the program can be closed. This is exactly what snapshots are for. The Take Snapshot tool is accessed from the Scenario menu.

When HexSim takes a snapshot of a scenario, it saves a copy without first subjecting it to error checking. To keep this snapshot from being used in a simulation, HexSim appends a tilde (~) to the end of the file name.



In the Workspace tab, snapshots are indicated with an asterisk to the left of the scenario name. In addition, the file date is not displayed for snapshots. Finally, the scenario

name will be prefixed with an asterisk in the scenario tab, when one is opened. See the image above for illustrations of each of these visual cues.

A snapshot may exist on its own, or it may reside side-by-side with a valid scenario. If both a valid scenario and a snapshot are present, the snapshot will always be loaded.

The Workspace Scenarios context menu for snapshots includes an item called Revert to Saved Scenario (see figure above). If this is selected, the snapshot will be deleted, and the saved scenario will be loaded.

If any errors in a snapshot are corrected, and it is saved, it will replace any other saved scenario. After the save, the snapshot will be gone.

Snapshots can be created two different ways. The preferred method is to select Take Snapshot from the Scenario drop-down menu. But as a precaution, if a scenario is open, but contains errors, and then HexSim itself is closed, a snapshot will be produced. This will not happen if the scenario has been closed prior to exiting HexSim.

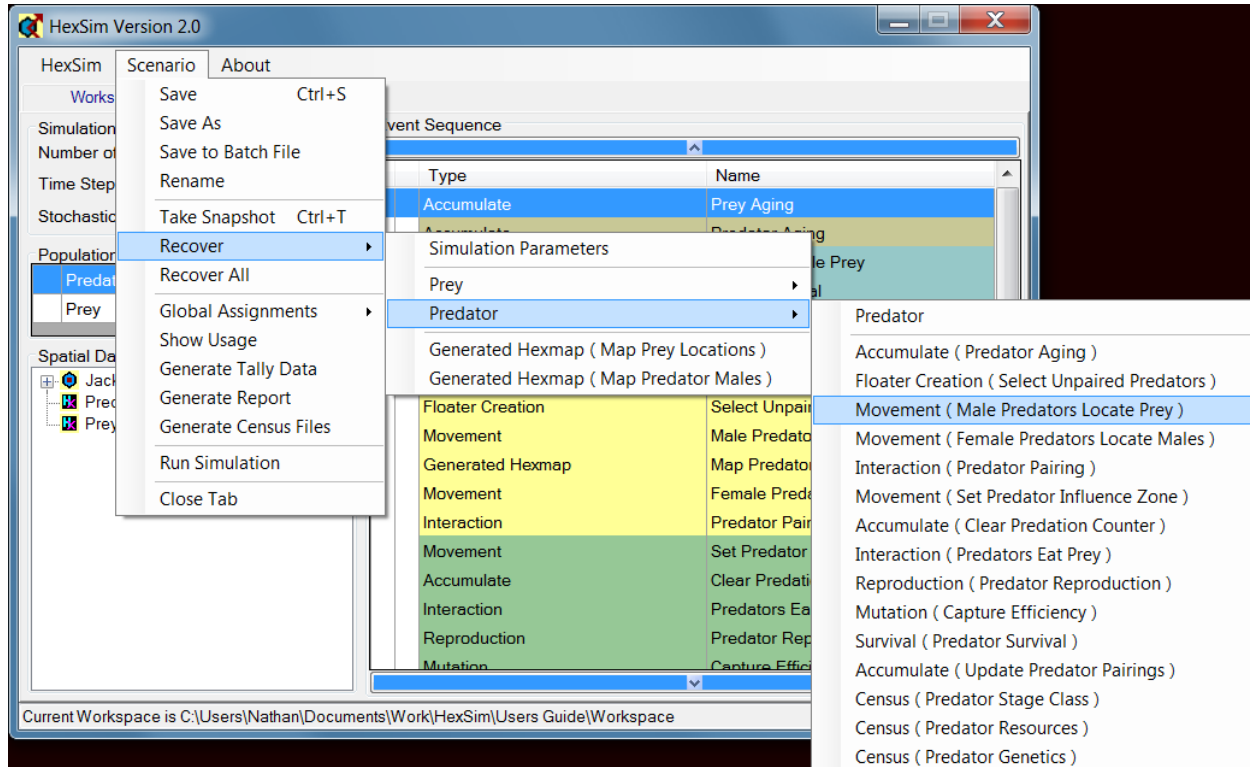
If a scenario is deleted, and that scenario includes a snapshot, then the snapshot will be deleted as well.

Using Recover

HexSim's Scenario drop-down menu includes a Recover and a Recover All option. Recover allows users to restore the parameter values associated with specific component of a simulation from the data stored on the hard drive. The Recover All option reloads the entire collection of scenario parameters at once, and is functionally equivalent to using the Reload from Saved Scenario context menu item associated with scenarios in the Workspace tab.

The Recover menu item has two submenus (see figure below). The first allows users to select the Simulation Parameters (Number of Replicates, Time Steps / Replicate, and Stochasticity), and any Generated HexMap events that have been placed into the scenario. None of these parameters are specific to a single population. Finally, this submenu allows users to navigate to an additional sub-menu for each population present in the scenario.

The population-specific sub-menus allow users to select and reload the Population Parameters (the top-most item). The sub-menu also allows users to reload the saved parameters for any event that has been assigned to that population.

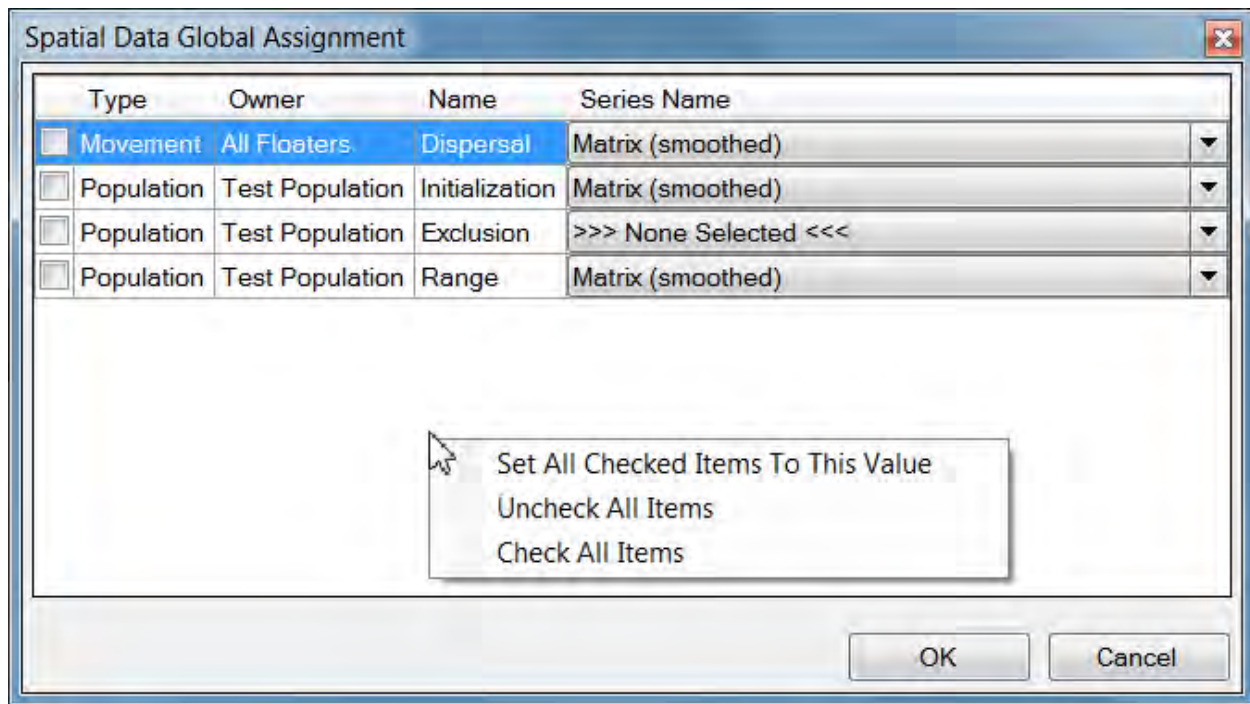


Global Assignment

Sometimes many changes must be made to a scenario all at once. For example, it might become necessary to substitute one spatial data time series for another throughout a scenario. If many events use this spatial data, then doing so one parameterization window at a time can be tedious and time consuming. The Global Assignment tools each allow users to change one type of data throughout a simulation, using a single dialog window. Global Assignment tools are available for spatial data (HexMap time series), barrier time series, affinities, log parameters, and descriptions. Each tool has commonalities and differences. But they can each add greatly to the efficiency with which a scenario can be edited. They are described below. The Global Assignment tool is accessed from the Scenario menu.

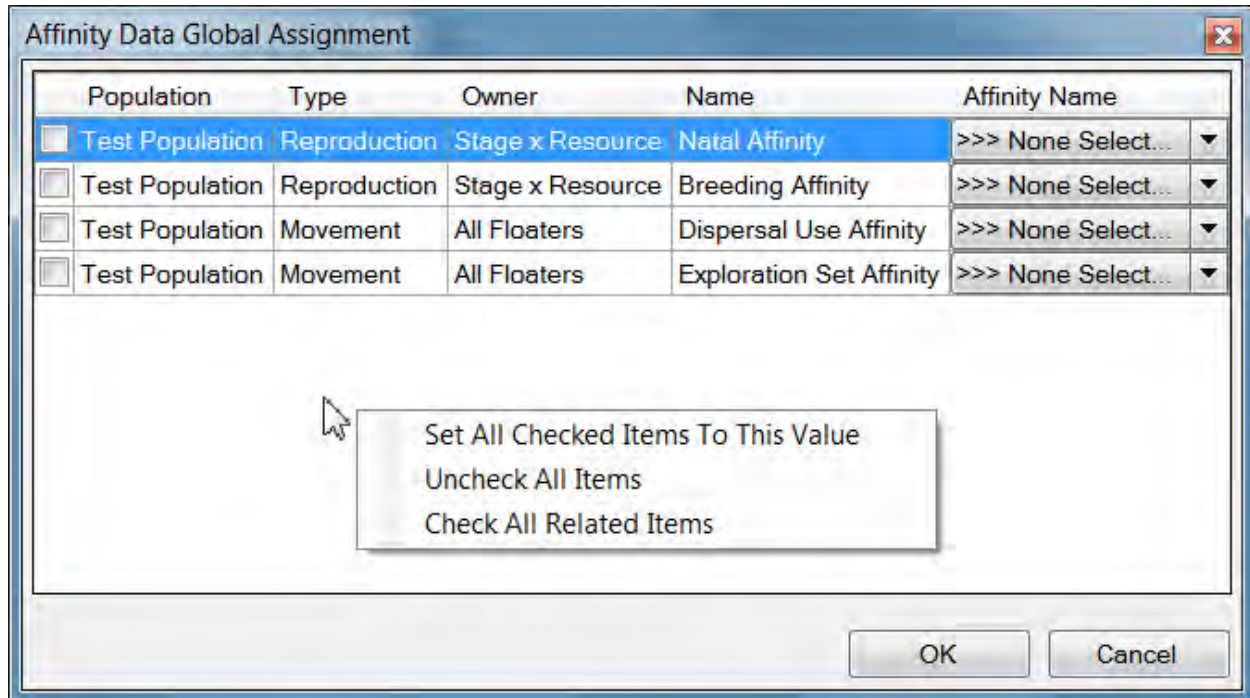
Spatial Data and Barriers

Both the Spatial Data (HexMap time series) and Barrier Time Series global assignment tools use exactly the same interface. Each component of a scenario that includes spatial data or barrier data is included as a separate row. The drop-down menus on the right allow users to assign the desired spatial data or barrier data to each item that must be altered. A context menu item exists for setting all checked items to a single selected value.



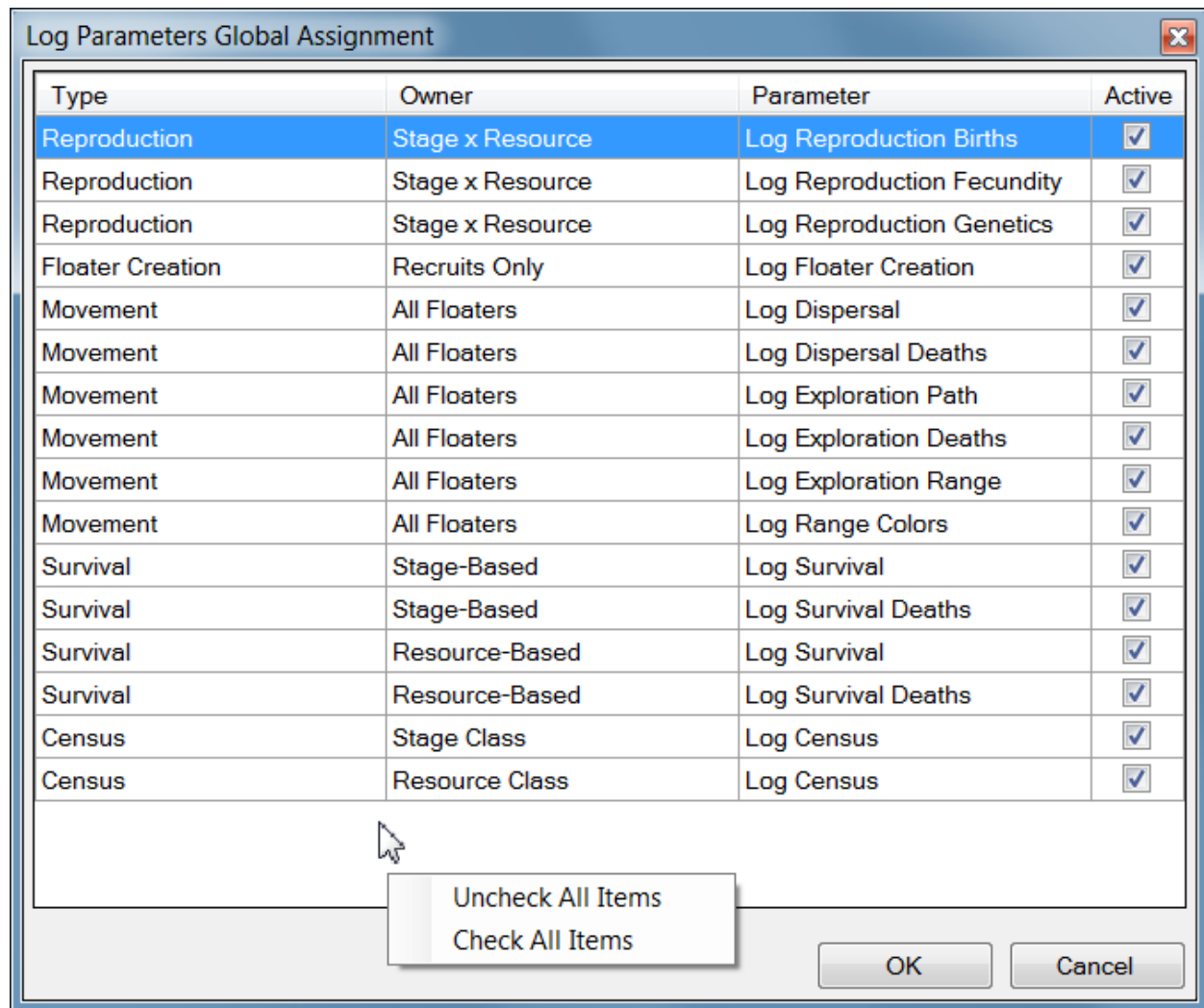
Affinities

The Affinity Data global assignment tool contains a separate row for each event that uses affinity data. The drop-down menus on the right allow users to assign the desired affinity data to each item that must be altered. A context menu item exists for setting all checked items to a single selected value.



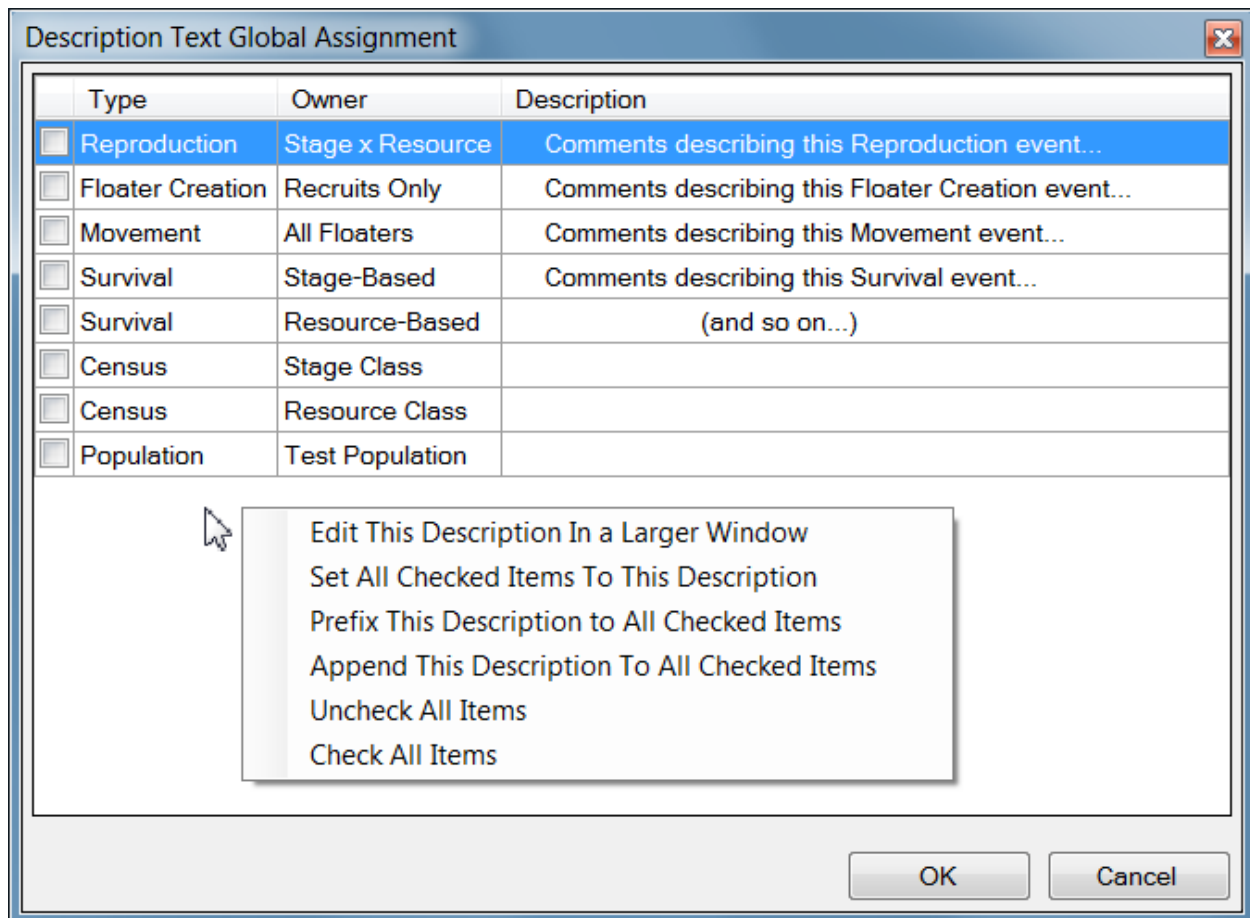
Log Parameters

The Log Parameters global assignment tool contains a separate row for every event's logging options. The check-boxes on the right allow users to quickly turn logging on or off in each of these instances.



Descriptions

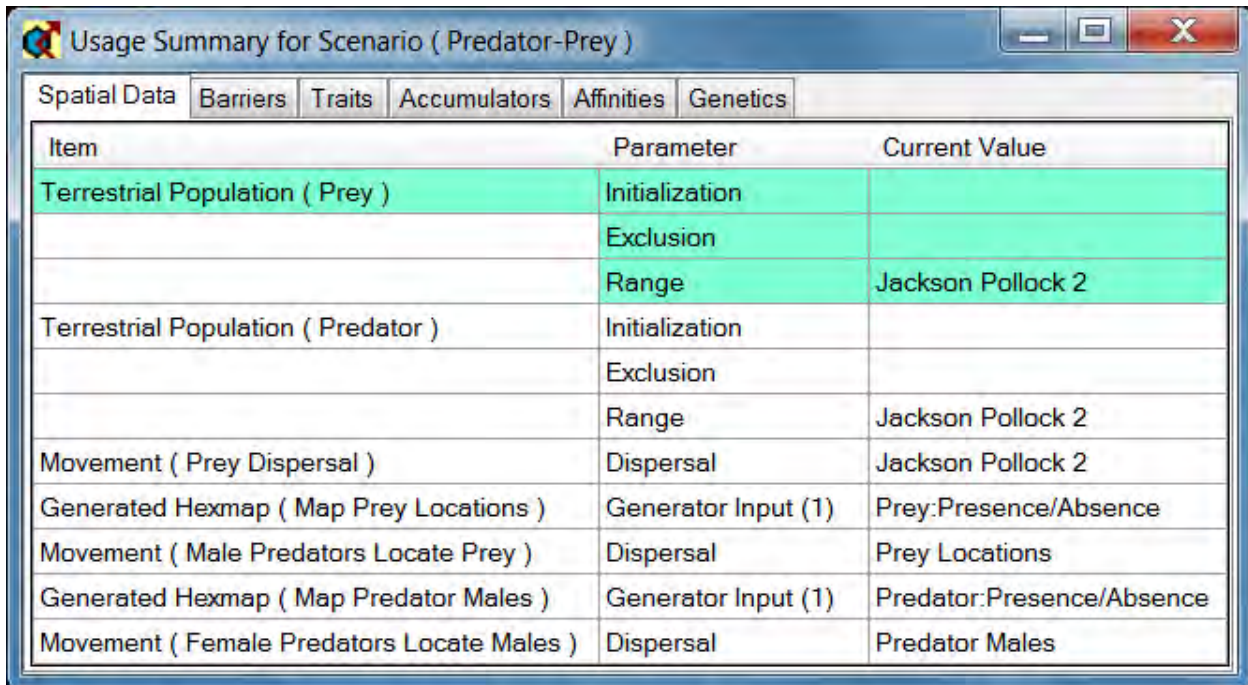
The Descriptions global assignment tool contains a separate row for every description tab present in the scenario. Users can edit the description text from within the tool, and the results will be propagated to the appropriate tab. Several context menu items are provided to facilitate the process. Users may edit the description text in a larger editing window. They may assign a single description to all checked items. There are also options for prefixing and appending new text to existing descriptions.



Show Usage

Scenarios can become large and complex. To help manage complex scenarios, a Usage Summary tool is included with HexSim. This tool provides users with a summary of all of the important dependencies they have built into a scenario. Users can see every use of a particular spatial data time series, or barrier series. They can see which traits and trait values have been used to parameterize populations and events, and what uses have been made of a population's accumulators. Event's use of affinities and genetic traits can be displayed as well.

The Usage Summary dialog window can be accessed from the Scenario menu by selecting the Show Usage option. The Usage Summary has separate tabs for Spatial Data, Barriers, Traits, Accumulators, Affinities, and Genetics. An image of the Usage Summary window is shown below.

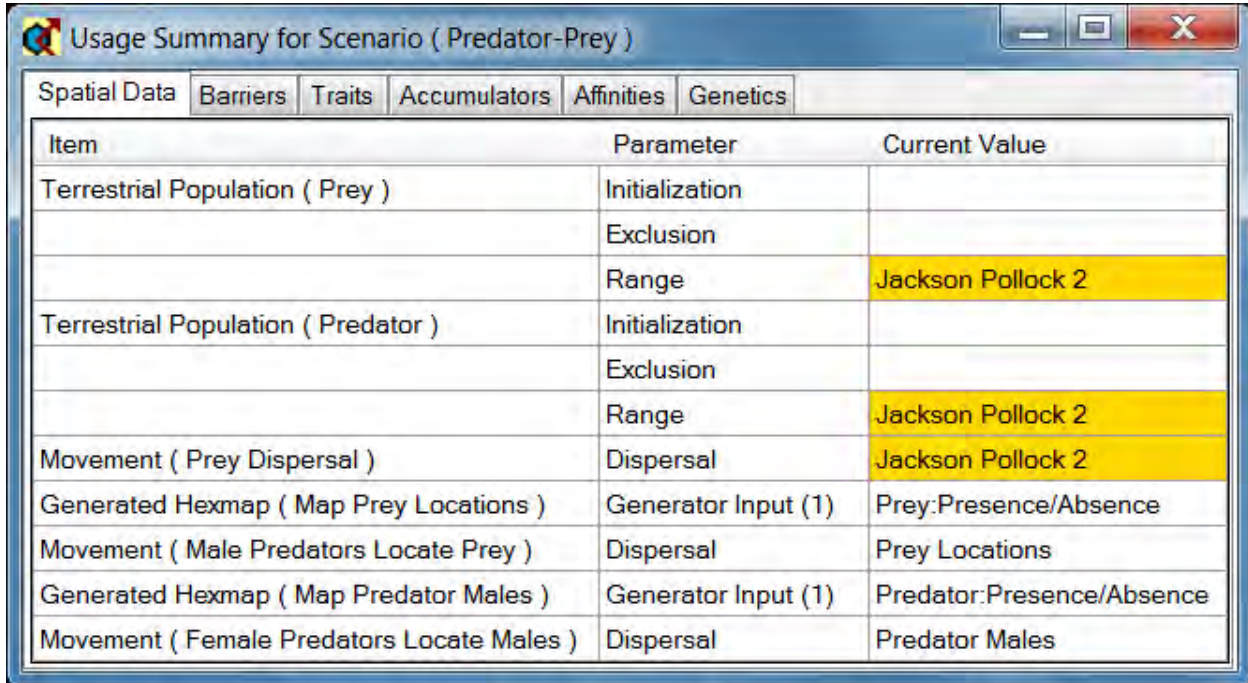


Item	Parameter	Current Value
Terrestrial Population (Prey)	Initialization	
	Exclusion	
	Range	Jackson Pollock 2
Terrestrial Population (Predator)	Initialization	
	Exclusion	
	Range	Jackson Pollock 2
Movement (Prey Dispersal)	Dispersal	Jackson Pollock 2
Generated Hexmap (Map Prey Locations)	Generator Input (1)	Prey:Presence/Absence
Movement (Male Predators Locate Prey)	Dispersal	Prey Locations
Generated Hexmap (Map Predator Males)	Generator Input (1)	Predator:Presence/Absence
Movement (Female Predators Locate Males)	Dispersal	Predator Males

Each Usage Summary tab has three columns labeled Item, Parameter, and Current Value. Items refer to parameterization windows or events. Parameters refer to specific parameters contained within the windows or events. Current Value refers to the value assigned to a parameter.

The cells in the Usage Summary window can be selected by clicking on them. When a cell in the Item or Parameter columns is selected, the items associated with it are highlighted. So for example, when an cell in the Item column is selected, all of the

parameters and current values associated with that item will be highlighted in green. The green highlighting helps users visualize the groupings between events, parameters, and values. However, when a cell in the Current Value column is selected, all of the cells with the same value are highlighted, in orange. The orange highlighting helps users see where in a scenario a specific parameter value (e.g. a specific spatial data time series) has been used (see figure below).



Item	Parameter	Current Value
Terrestrial Population (Prey)	Initialization	
	Exclusion	
	Range	Jackson Pollock 2
Terrestrial Population (Predator)	Initialization	
	Exclusion	
	Range	Jackson Pollock 2
Movement (Prey Dispersal)	Dispersal	Jackson Pollock 2
Generated Hexmap (Map Prey Locations)	Generator Input (1)	Prey:Presence/Absence
Movement (Male Predators Locate Prey)	Dispersal	Prey Locations
Generated Hexmap (Map Predator Males)	Generator Input (1)	Predator:Presence/Absence
Movement (Female Predators Locate Males)	Dispersal	Predator Males

The populations and events context menus include options to Show Usage for the Selected Population, and Show Usage for the Selected Event. These menu items also call up the Usage Summary tool, but in these cases it will only load the relevant usage data.

Importing Data

The HexSim drop-down menu has options for importing scenarios, HexMaps, and Barrier Maps. Each of these options are discussed below.

Import Scenario

This option copies a scenario and places the copy in the current workspace. The source will normally be a scenario XML file in another workspace's Scenarios folder. The Import Scenario tool simply calls up a file chooser that lets users select the scenario to be imported.

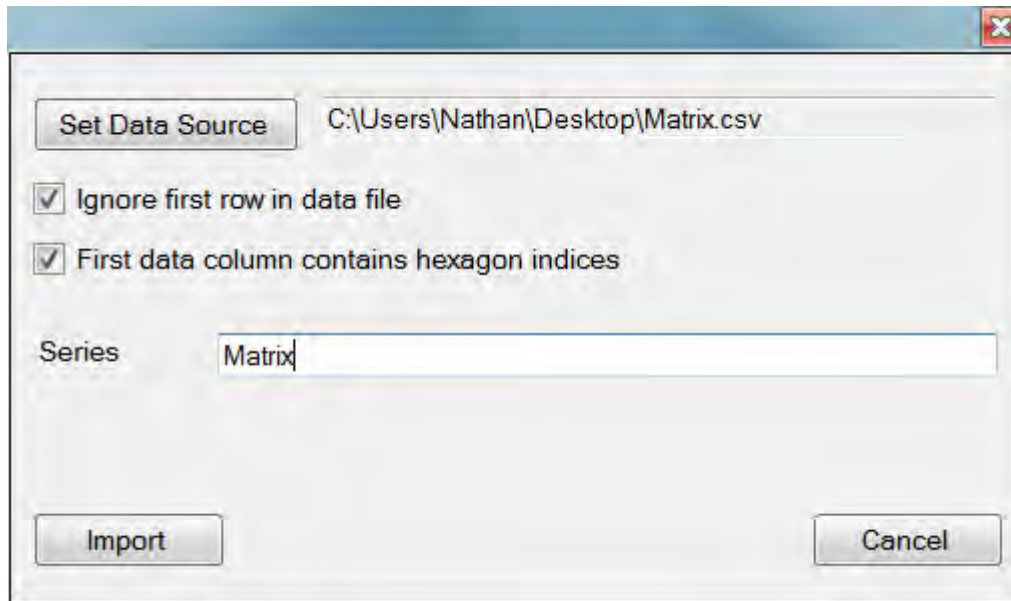
Import HexMap Data

When importing HexMap data, you can read DBF, CSV, and TXT file types.

The requisite file formats for CSV and TXT files are exactly the same. Both must be comma-separated text files. And in both cases white-space is OK. DBF files are assumed to have originated as part of an ESRI Shapefile.

The import window allows includes a setting for specifying whether the file has a header row. The heading, if one exists, must be the first row in the file. No accommodation is made for multi-row headers. The tool also has a setting for specifying whether the first column lists the hexagon indices. If the hexagon indices are not included in the file, then they are inferred from the row number (counting from 1 up, as there is no hexagon 0).

Finally, a series name must be specified. The name selected must not match that of any spatial data series already present in the workspace. An image of the Import HexMap dialog window is shown below.



If multiple data columns exist in the imported file, then they will be used to construct multiple time steps. The first data column will be used to construct time step 1, the second will be used to construct time step 2, and so on. The Workspace Spatial Data context menu item called Rename Timestep can be used to change the time step values after the import has completed.

Finally, when importing HexMap data, it's not necessary that there be exactly one value for each hexagon. Hexagons for which there is no data will simply be scored zero. Thus sparse HexMaps can be developed by building a file with entries for only the hexagons with non-zero values. In such cases, a column of hexagon indices will typically be necessary. Also, imported files are read from top to bottom, and duplicate entries are allowed. If multiple entries exist for a single hexagon, the last entry is the one that will end up being used. This makes it possible for users to export a HexMap as a CSV file, append changes to the bottom of the file, and then import the edited file.

Importing Barrier Data

Barrier data can be imported from SHP and HBF files. SHP files are one component of an ESRI Shapefile. HBF files are used to store HexSim Barrier Maps, and the extension stands for "HexSim Barrier Format". ESRI Shapefiles can hold polygon, line, or point data. Only SHP files holding line data can be imported into HexSim to create a barrier map. When importing HBF data, the edge data is simply copied hexagon-by-hexagon. If HexSim encounters a hexagon ID that exceeds the number of hexagons in the current workspace, an error is generated and the import is aborted. When importing a SHP file, HexSim snaps each line segment in the file to a series of hexagon edges. For every hexagon a line intersects, one or more hexagon edges are selected that most closely approximate the path taken by the line through the hexagon. A pair of barrier edges are then placed at each side of the hexagon to which the line segment was snapped.

Like HexMaps, Barrier maps are also time series. But a single barrier series can have multiple barrier category pairs. Barrier data come in pairs because each side of a HexSim barrier can exhibit different properties. Individuals interact with the barrier edge they cross when they exit a hexagon. If a Barrier map has multiple barrier category pairs, then each side of each pair can have unique properties. HexSim barriers have probabilities for Mortality, Deflection, and Transmission. Transmission is automatically set to $1.0 - (\text{Mortality} + \text{Deflection})$. These probabilities are not specified during the import process. When importing barrier data, the series name and time step must be specified. HexSim will set the barrier category pair ID values automatically. These are displayed in the interface as the Primary and Companion Categories. The first pair are categories 1 & 2. Next are 3 & 4, and so on. In contrast to the import of HexMap data, it is not possible to import multiple time steps of barrier data from a single file.

When a SHP file is selected for import, a pair of parameters are added to the import dialog window. These parameters allow users to temporarily translate the Shapefile data so that it is coincident with the HexSim grid. The south-west corner of HexSim's grids are always set to (0, 0) meters. The south-west corner of the Shapefile data is very unlikely to lie at (0, 0) meters. The West Edge and South Edge parameters should be set equal to the west and south edges of the coverage that the Shapefile data was extracted from. These values will be subtracted from the Shapefile line coordinates. An image of the barrier import dialog window is displayed below.

Set Import Source and Destination Barrier

Set Data Source C:\Users\Nathan\Desktop\NSO Modeling Region Boundaries.shp

Series Name: NSO Modeling Region Boundaries Series Step: 1

West Edge (in meters) 225030.0000 New Primary Category 1

South Edge (in meters) 424980.0000 New Companion Category 2

Import Cancel

After a SHP file is imported, a summary dialog window will be displayed. This window shows the grid coordinates, the extent of the Shapefile line data (prior to translation by the West Edge and South Edge parameters), and the number of line segments that were converted into barriers. An image of this summary window is displayed below.



Some clean-up may be required after barrier import. Hexagon edges can end up being specified multiple times, but with opposite orientations. Conflicts will be displayed in magenta, and they must be hand-edited out of the barrier data.

Batch Processing

HexSim includes a facility for automatically running simulations for collections of scenarios. Users develop a list of scenarios to run, which is done from the HexSim interface. Then the BatchRunner.exe program is either run indirectly from a command window, or directly by double-clicking it. For each scenario in the list, BatchRunner calls the HexSim engine and passes it the appropriate parameters. Each instance of BatchRunner will work within just a single workspace. But multiple instances of BatchRunner may be launched at the same time. BatchRunner.exe is one of the files that makes up the HexSim model.

In order to use BatchRunner, users must first create a batch file. Only a single batch file can exist per workspace. The batch file lives at the top of a workspace, and will appear next to the grid file. Batch files are always named "batchfile.xml". There are two tools available for creating and maintaining batch files. First, the HexSim Scenario menu has a Save to Batch File option. This will add the current scenario to the collection that will be processed by BatchRunner. Second, the HexSim menu has an Edit the Batch File option. This calls up the Batch List editor, which is shown below.



The Batch List editor can be used to add, remove, and reorder scenarios. This tool has a context menu that allows users to:

- Add a Scenario
- Add All Available Scenarios
- Remove the Selected Scenario
- Remove all Scenarios

BatchRunner will run the scenarios specified in the Batch List tool, in the order that they appear. A single simulation will be generated for each scenario, regardless of the number of replicates specified in the scenario. That is, BatchRunner does not break an N-replicate scenario into N single-replicate simulations.

BatchRunner takes two arguments. The first is the number of simulations to run simultaneously. The number of simultaneous simulations defaults to 1. The second is the path to the batch file. If BatchRunner is launched by double-clicking the file, then the number of simultaneous simulations will be fixed at 1. If the computer being used has a multiple processors, then it may be advantageous to launch BatchRunner from a

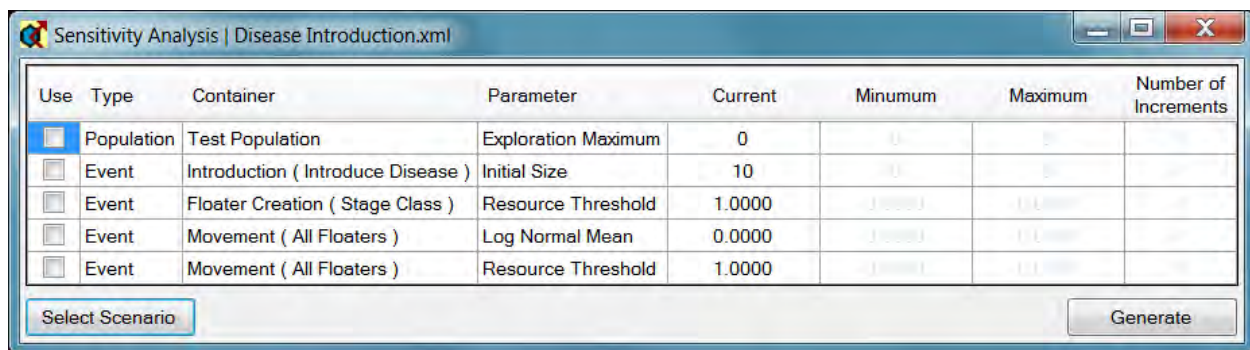
command window and supply a value exceeding 1 for the number of simultaneous simulations. When this is done, BatchRunner will keep that number of simulations running at all times, until the list has been exhausted. To summarize, BatchRunner is called as follows:

- `BatchRunner.exe N <path-to-batch-file>`
Runs N simultaneous simulations using the batch file specified by <path-to-batch-file>
- `BatchRunner.exe <path-to-batch-file>`
Runs the simulations in the batch file specified by <path-to-batch-file> one at a time.
- `BatchRunner.exe N`
Calls up a file chooser so that the batch file may be selected graphically. Runs N simultaneous simulations using this batch file.
- `BatchRunner.exe <path-to-batch-file>`
Calls up a file chooser so that the batch file may be selected graphically. Runs the simulations listed in this batch file one at a time.

If BatchRunner is launched by double-clicking the file, or if it is run from a command window without including a path to a batch file, then a file-chooser dialog window will come up. This dialog window is used to select the batch file that will be loaded. When BatchRunner is run from a command window, it will send feedback on simulation progress back to that window. The only way to kill the simulations that BatchRunner starts is to use the Windows Task Manager. Look for processes named HexSimEngine.exe.

Sensitivity Analysis

HexSim includes a tool that facilitates sensitivity analyses. This feature is an extension of HexSim's [Batch Processing](#) module. The Sensitivity Analysis tool is located in the HexSim release folder. Users double-click the SensitivityAnalysis.exe file to start the tool. Once the Sensitivity Analysis interface has started, a scenario must be selected using the Select Scenario button in the lower-left corner. Scenarios can be selected from any workspace. Once a scenario has been read, a subset of its parameters will be displayed in the tool. Only real-valued and integer-valued parameters are accessible in the tool. Categorical parameters cannot be used. Not all numeric parameters are available in the Sensitivity Analysis tool. Including every parameter would unnecessary clutter the interface. Available parameters can be selected by clicking the check-boxes on the far left. For each selected parameter, a Minimum, Maximum, and a Number of increments must be supplied. These values are used to create a series of parameter increments. An image of the Sensitivity Analysis tool is shown below:



HexSim's Sensitivity Analysis tool creates a series of scenarios that capture the desired variation in one or more parameters. If multiple parameters are selected, a series of increments will be computed for each. When the Generate button is pushed, a HexSim scenario will be created for each possible permutation of every parameter increment. These scenarios will be placed in a "Sensitivity Analysis" folder at the top of the workspace holding the selected scenario. If a Sensitivity Analysis folder already exists at the time that the Generate button is pushed, its existing content will be deleted. A XML file called "batchfile.xml" will be written to the Sensitivity Analysis folder. BatchRunner.exe (see the section on Batch Processing) can use batchfile.xml to automatically run each of the scenarios. The generated scenario files are named using a combination of the selected scenario name and a numeric index. The parameter values in a generated scenario cannot be inferred from its filename. But a file named KEY.csv is included in the Sensitivity Analysis folder, and this file associates the model parameter values with the scenario filenames. KEY.csv can be opened in a spreadsheet or text editor. Users can also open the scenario files directly to examine their content. However, they must first be moved or copied to the workspace Scenarios folder. However, if a scenario is moved from the Sensitivity Analysis folder to the Scenarios

folder, then BatchRunner will not be able to locate it when its run with the sensitivity analysis batchfile.

Increment generation is performed slightly differently depending on whether a parameter is real or integer-valued. Let X and Y represent the Minimum and Maximum settings, respectively. Let N represent the Number of Increments, and let S represent the increment size. For real-valued parameters, the increments generated will be:

$$\{ X, X + S, X + 2S, \dots, Y \}$$

where $S = (Y - X) \div (N - 1)$. For integer-valued parameters, the increments generated will be:

$$\{ X, X + S, X + 2S, \dots, X + (N - 1)S \}$$

where S is computed as above, but then truncated to produce an integer. For integer parameters, the Number of Increments cannot be set to a value that produces a S less than 1.0. The Maximum may not be reached when working with integer parameters.

The elements that are available to SensitivityAnalysis.exe are a subset of the numeric parameters from HexSim populations and events. Many of HexSim's parameters are linked through dependencies. For example, the Movement Event's Dispersal tab has values for Repulsion Minimum and Maximum. The acceptable values for Repulsion Minimum change depending on what the maximum has been set to. In the Exploration tab of Movement events, the Resource Threshold parameter is only meaningful when the Exploration Goal includes joining a group. So it would not make sense to evaluate the sensitivity of a HexSim simulation to the value of the Resource Threshold if the Exploration Goal was set to Start a New Group. Changes to the Maximum Range Area can invalidate the Maximum Range Span parameter. Competitive Ability is only used when its check-box is checked. In addition, survival, reproduction, transition, and mutation rates all come in tables, and whole families of these rate tables can be assigned to any one event in order to simulate environmental stochasticity. This structure makes these parameter values incompatible with the sensitivity analysis program. All of these, and other complications limited the number of parameters that could be modified using HexSim's sensitivity analysis utility.

The parameters that can be varied using SensitivityAnalysis.exe are listed in the table below:

CONTAINER	PARAMETER
Population	Initial Population Size
	Maximum Group Members
	Hexagons Range-Eligible if Value is At Least
	Minimum Range Resources
	Floater Preemption of Group Resources
Floater Creation	Resource Threshold

Introduction	Individuals
Movement	Log Normal Mean
	Log Normal Standard Deviation
	Path Length Minimum
	Path Length Maximum
	Percent Auto-Correlation
	Trend Period
	Maximum Explored Area
Set Group Affinity	Log Normal Mean
	Log Normal Standard Deviation
	Path Length Minimum
	Path Length Maximum
	Percent Auto-Correlation
	Trend Period
	Resource Target
	Maximum Explored Area

Real-valued parameters that are treated as a percentage in the interface will be stored as a value between 0 and 1. If users want to vary other parameters not in this table, this can be easily done by hand. The section on Batch Processing describes how to accomplish this.

If the scenario being used includes a Movement event with the strategy set to Translocate then Explore, dispersal parameters will appear in the sensitivity analysis tool that are not visible in the interface, and are not used. These parameters should not be selected.

Command Line Options

HexSimEngine

HexSimEngine.exe and HexSimEngine64.exe are the HexSim model engine executables. Depending on the computer being used, one of these is launched by the model interface when a simulation is started. But both can be run from a command line as well. The syntax used for these programs is

HexSimEngine -d -r seed scenario

The *seed* is a number used to initialize the random number generator, and *scenario* represents the full path to a scenario file. The -d and -r flags are both optional, and they are used as follows:

Value	Meaning
-d	Indicates that a time-stamped results directory should be used.
-r	Sets the random number seed. If this flag is unused, a random seed will be generated.

BatchRunner

BatchRunner.exe will run a collection of scenarios in series or parallel. The syntax for BatchRunner is

BatchRunner n batch_file

The optional parameter *n* specifies the number of simulations to run simultaneously. The number of simultaneous simulations defaults to 1 if *n* is not supplied. The *batch_file* parameter is the full path to a workspace's batch file. If *batch_file* is not supplied, then a file chooser will come up, allowing the user to select a batch file graphically.

OutputTransformer

OutputTransformer.exe can create reports or tallies. Normally, OutputTransformer is run from the model interface when reports or tallies are selected from the Scenario menu. But OutputTransformer can be run from a command line as well. The syntax for OutputTransformer is described below.

For reports:

OutputTransformer.exe -type:start-end logfile

where **type** can be:

Value	Meaning
births	Births
deaths	Deaths
genetics	Genetics
vitals	Vitals
movement	Movement
ranges	Ranges
barriers	Barriers
matrices	Stochasticity
interactions	Interactions
dispersalSteps	Dispersal Steps
dispersalSuccesses	Dispersal Successes
dispersalPaths	Dispersal Paths
projectionMatrix	Projection Matrix

start is the first time step to be processed (defaults to 0),

end is the last time step to be processed (defaults to the last step in the simulation),

and **logfile** is the full path to the log file being used.

For example, to create a births report for steps 0 - 10 using data found in a log file named test.log, the command issued would be:

OutputTransformer.exe -births:0-10 test.log

For tallies:

OutputTransformer.exe -format:type:rows:cols:narrow:start-end:groups,floaters logfile

where **format** can be either "csv" or "hexmap",

type can be:

Value	Meaning
b	Births
d	Deaths

n	Births – Deaths
x	Barrier Deaths
y	Barrier Deflections
z	Barrier Transmissions
o	Dispersal Flux
i	Interactions
u	Occupancy

rows is the number of rows in the grid,

cols is the number of columns in the grid,

narrow can be either "true" or "false",

start is the first time step to be processed (defaults to 0),

end is the last time step to be processed (defaults to the last step in the simulation),

groups can be either "true" or "false",

floaters can be either "true" or "false",

and **logfile** is the full path to the log file being used.

For example, to create a CSV tally of births on a 100 x 100 narrow grid for steps 0 - 10 using data on only groups, found in a log file named test.log, the command issued would be:

```
OutputTransformer.exe -csv:b:100:100:true:0:10:true,false
test.log
```

It is possible to generate an occupancy tally stratified by trait combinations from the command line. But the list of additional arguments that must be passed is so complex that this operation should only be performed using the HexSim interface.

HexMapConverter

HexMapConverter.exe can be used to convert CSV and DBF files into HexSim spatial data time series. Each CSV or DBF file may contain one or more data columns. These data columns will be converted into successive time steps in the spatial data time series. The CSV and DBF files may also contain option headers and a column of hexagon IDs. Headers, if present, must be limited to a single line. If a column of hexagon IDs is present, it must be the first (left-most) column in the file.

HexMapConverter.exe is called with the following command-line arguments

HexMapConverter.exe Input Header Hex_IDs Rows Columns Narrow Output

The command-line arguments are explained further in the table below

Value	Meaning
Input	This is the full path to either a CSV or a DBF file.
Header	This is either the string "true" or "false", depending on whether a header is present in the file. Do not include the quotation marks surrounding "true" or "false".
Hex_IDs	This is either the string "true" or "false", depending on whether the first column in the file contains a list of hexagon IDs. If this is set to false (no hexagon IDs), then hexagon ID is inferred from row number, beginning with number 1. Do not include the quotation marks surrounding "true" or "false".
Rows	The number of rows in the grid that the spatial data time series will be used with.
Columns	The number of columns in the grid that the spatial data time series will be used with. If the grid is narrow, then even numbered rows will have one less hexagon. Thus this value should be set to the maximum number of columns present in the grid.
Narrow	This is either the string "true" or "false", depending on whether the grid is wide or narrow. In wide grids, every row contains the same number of hexagons. In narrow grids, even rows contain one less hexagon than wide rows. Do not include the quotation marks surrounding "true" or "false".
Output	This is the full path to the folder in which the individual HXN files will be written. If this folder does not yet exist, it will be created.

For example, to convert a csv file named test into a HexMap named demo, assuming a wide grid with 50 columns and 100 rows, and assuming the csv file contains a header and hexagon IDs, the command would be:

```
HexMapConverter.exe test.csv true true 100 50 false demo
```

But the full path names to HexMapConverter.exe, test.csv and demo would all have to be provided.

Any missing rows in the input file will be converted to zeros in the output. If hexagons are specified multiple times in the input, the output will contain the last value.

Troubleshooting

Troubleshooting

[Common Mistakes](#)

[Model Calibration](#)

[Model Performance](#)

Common Mistakes

Many types of errors can be unintentionally built into valid HexSim scenarios. This section will list some common mistakes, but it is not exhaustive. The descriptions here are brief, and they assume familiarity with the model.

Populations

Initial Population Placement

If the initial population is placed randomly, they may die before they get a chance to acquire the resources necessary to generate a positive growth rate.

Initial Population Resources

Make sure that the initial population has a chance to acquire resources before subjecting them to a Survival event that has been stratified by a resource acquisition trait.

Exclusion Series

It is easy to unintentionally exclude populations from the resources they need by setting up the Exclusion Series parameter incorrectly.

Range Data

Ranges will be difficult or impossible to construct if the Maximum Range Span, Range-Eligibility, or Minimum Range Resources parameters are poorly selected. Since only group members can reproduce, this can have a huge impact on population growth rate.

Range Barriers

Range Dynamics events, and immigration into groups can cause ranges to change shape and size. If impenetrable movement barriers are used, but range barriers are not set, then these events can cause what appears to be prohibited barrier-crossings.

Resource Acquisition

The four resource acquisition updater functions set accumulators to a value between 0 and 100%. If accumulated traits are developed that use such an

accumulator, the thresholds must be set properly. Using the resource acquisition trait builder can help avoid mistakes.

Competition

It is easy to set up an Accumulate event to track resource competition between two or more populations. The two mechanisms for doing so involve use of the Competitive Acquired or Competitive Explored Updater functions. If users develop such an Accumulate event, but forget to set up each population's Competitive Ability parameters, then the populations will fail to compete. The Competitive Ability parameters are located in the Population Parameters Range Data tab.

Event Sequence

Floater Creation Events

Users will often forget to include a Floater Creation Event in the event sequence. This can have strange and dramatic impacts on the population dynamics. Without a floater creation event, recruits will never leave the natal site, and the population will usually decline rapidly.

When a scenario uses a stage class trait, recruits will typically start life as stage-0 individuals. Often some or all of the recruits will be forced by a Floater Creation event to leave their natal group. Problems will arise if this Floater Creation event is preceded by an Accumulate event that makes individuals older. If the Accumulate event moves the recruits into the next stage class, then there will be nobody for the Floater Creation event to act upon. This problem can be easy to overlook when debugging a scenario. To avoid this issue, make sure that Floater Creation is called after Reproduction, but before individuals age.

Accumulate and Transition Events

It is easy to develop multiple accumulated and probabilistic traits, and to subsequently use them to stratify life history events, such as survival and reproduction. Users may forget to include Accumulate and Transition events in the life cycle to maintain individual accumulated and probabilistic trait values. This will not cause errors, as every individual will be assigned a valid set of trait values when it is created. But leaving out Accumulate and Transition events will eliminate feedbacks that form essential components of the simulation.

Resource Acquisition

Nothing special must be done to successfully use the Acquired Resources updater function to maintain a resource acquisition trait. However, if the Competitive Acquired Resources updater function is used, then at least two populations should

have non-zero Competitive Ability parameters set in their Population Parameters Range Data tab.

If the Explored Resources or Competitive Explored Resources updater functions are used, then the event sequence must include at least one Movement event with the Strategy parameter set to Construct Explored Areas. Also, at least one such Movement event must have the Clear Explored Counts check-box checked. This type of Movement event is needed to build the data tables that are used to quantify the resources available in individual's explored areas. If the Competitive Explored Resources updater function is being used, then this special type of Movement event must be present for each competing population. Each competing population must also have a non-zero Competitive Ability parameter set in its Population Parameters Range Data tab.

Event Order

Events in HexSim are processed sequentially from the top to the bottom of the event list. Altering the event order can have profound effects on the simulated population's dynamics. If survival precedes reproduction, then potential parents may die before they reproduce. If reproduction proceeds survival, then users should be careful not to double-count first year mortality by including it in both the survival rates and fecundities. If resource-dependent survival is preceded by movement, an Accumulate event might need to be placed in between so that individual's resource acquisition traits can be updated prior to survival.

Census Events

Census events gather data at the time that they run. If they are not placed correctly, then the information they gather might not be what was expected. For example, an Accumulate event dedicated to aging individuals might move all stage-zero individuals into stage class one. A Reproduction event would subsequently replenish the stage zero class. A census placed between these two events would reveal no individuals in stage class zero. Users must recognize that this is simply an effect of the timing of the census. A census placed after the Reproduction event would produce different results.

Events

Accumulate

Accumulate events can (among others) use four resource acquisition updater functions, a Quantify Environment updater, and with an Individual Locations updater. These updater functions are in some ways similar, but they are not interchangeable. Users should be familiar with their applications.

The resource acquisition updater functions return the percentage of an individual's target resource that they have been able to acquire. The Quantify Environment updater function measures exposure to some spatially-distributed quantity. The Individual Locations updater has special features that make it uniquely useful for locating individuals within the grid.

If the Explored Resources or Competitive Explored Resources updater functions are used, then at least one Movement event must use the Construct Explored Areas strategy. The Construct Explored Areas strategy creates the Explored Counts data structure that is used to divide hexagon resources up among multiple individuals. If no such Movement event exists, then each individual will have access to all of the resources in its explored area, regardless of the extent to which explored areas overlap.

Movement

If a Movement event uses the Construct Explored Areas strategy, then it should have the Clear Explored Counts check-box checked. If multiple Movement events use the Construct Explored Areas strategy, then the first one should have this box checked. The others should have the Clear Explored Counts box unchecked. The Clear Explored Counts check-box resets the Explored Counts data structure to zero. If this is not done once per time step, then the allocation of resources from individual explored areas will not work correctly.

Floater Creation

The Floater Creation event has several key parameters that make it flexible. If these features are not used, it can end up functioning as a fairly blunt instrument. The Resource Threshold parameter can be used to ensure that group resources fall below a specified threshold before anyone is forced to leave. The Enforce Maximum Group Size parameter can be used by itself to enforce just density-dependent dispersal. Numeric priorities can be used to carefully remove individuals from a group based on competition for resources.

Generated HexMaps

Smoothing and rescaling generated HexMaps can be used in conjunction with dispersal's attraction and repulsion parameters to guide searchers towards mates, or prey, etc. If population data, such as the presence of prey items, is being tallied cumulatively, then users will need to keep track of two separate generated HexMaps. The first will be the cumulative tally. The second will be a smoothed, rescaled, copy of the cumulative tally. The latter HexMap will be used in dispersal. If the cumulative tally itself (instead of the copy) is repeatedly smoothed and rescaled, then it will start to lose its meaning.

Census

Census events gather data that are critical for the generation of tallies, and to the functioning of the Simulation Viewer. But if multiple Census events appear next to one-another in the event sequence, then it will be beneficial to turn off logging for all but one of these Census events. Doing so will help limit the size of the log files produced when the simulation runs. It will also speed up both the generation of tallies, and the display of the Simulation Viewer. When multiple sequential Census events write output to the log file, this data is completely redundant.

Event Outputs

In order to limit the size of the log files created as a simulation runs, users may be tempted to turn off logging for one or more life history events. Doing so may limit HexSim's ability to generate reports or tallies from the simulation output, and also may interfere with the working of the Simulation Viewer. Turning off logging can also result in misleading results being generated. The output generated by Census events is an exception. It will not be affected by other event's logging parameters. A potential alternative is for users to generate the requisite reports and tallies, and then zip the simulation log file. Zipped log files are typically much smaller than their uncompressed counterparts.

Order Rows by Selected Traits

HexSim's Survival, Reproduction, Transition, and Mutation events each make use of a rates tab. These rates tabs have a context menu item named Order Rows by Selected Traits. This menu item calls up a tool that allows users to reorder the trait combinations. For that specific event, the reordering will be permanent.

Users must be careful when using this reordering feature because reordering the traits does not change the order of data that has been entered into an event. If no rates have yet been entered into the event, then the traits can be reordered safely. If rates have already been supplied, then they should probably be either reentered by hand, or exported to a text file, reordered in a text editor, and imported back into the event.

Miscellaneous

Scenario Names

If a space is placed at the end of a scenario name, HexSim will be unable to use that scenario. For this to happen, the space has to be immediately before the ".xml" filename extension. The problem is that HexSim asks Windows to construct a results folder with the scenario name, but minus the ".xml". When Windows constructs the folder, it omits the trailing space. But HexSim tries to write its results to a folder with a name that includes the trailing space. This causes the simulation to fail.

Model Calibration

HexSim can be used to simulate real or hypothetical populations. When the target is an actual population, users will want to ensure that their scenario is as valid as possible. This is nontrivial, and especially so when the simulation model is spatially-explicit, individual-based, and feature-rich. HexSim is a framework within which models can be constructed. HexSim scenarios are the actual models. Thus for HexSim itself, validity comes down to making sure each feature actually accomplishes what is expected of it. This amounts to bug-detection and bug-fixing, and HexSim has been subjected to an extraordinary amount of such testing. In contrast, it is not possible to validate a HexSim scenario, or any model of a actual population, because all models are relatively simple abstractions of highly complex and poorly understood real-world phenomena. The best that can be done is to carefully calibrate a model's design and its parameters, and then to develop plausibility tests that compare simulated population trends and behavior to actual data. This section presents some approaches that may be useful for constructing well-calibrated HexSim scenarios that produce plausible results. This is not an exhaustive treatment, and this topic is not even an exact science. The most important thing users can do is to always construct the *simplest* plausible model. Adding complexity beyond that which is necessary to capture critical dynamics make models harder to understand and consequently less useful. However, its often only through trial and error that we determine how much complexity is absolutely necessary.

The tools HexSim users have for model calibration are the population trends captured by Census events, the quantifications of model dynamics obtained from tallies and reports, and the insights gathered from the Simulation Viewer. Census events are a principal tool because their estimates of population trends are easily compared to actual data. Also, Census events stratify populations by trait values, so they can provide insights into more subtle population dynamics, such as sex-ratios, resource-acquisition class sizes, or numbers of diseased individuals, etc. Tallies provide users with data on a population's use of a landscape. Tallies are used to generate maps, and thus they are particularly valuable for communicating simulation results to experts who are not themselves modelers. Tallies can be generated for specific temporal windows, so users may, for example, construct a time series of tallied maps illustrating decadal changes in population distribution. Reports provide users with more detailed understanding of a simulation dynamics. The Vitals report, for example, breaks all survival and reproduction events down by individual trait combinations, so users can see exactly how vital rates were influenced by trait values.

Users ability to calibrate HexSim simulations will depend on how much real population data is available. Data-rich environments encourage the construction of more detailed scenarios, and allow for more precise calibration. But to effectively calibrate a HexSim scenario, users will also have to become familiar with the tools discussed in the

previous paragraph. The breadth and depth of these tools should be adequate to meet users calibration needs.

Initial Population Size and Distribution

Start-up bias is a fundamental feature of all complex simulation models. Start-up bias results from discrepancies between a model's initial state and its steady-state conditions. While a simulation is settling into a steady-state, it will exhibit transient dynamics. Once a steady-state has been reached, the transient dynamics will die away. The initial population size and distribution are examples of parameters that users may purposely set far from steady-state at model initiation. For example, if a simulated population is declining in reality, it often makes sense to start simulations out with large numbers of individuals spread throughout the landscape. Then if the steady-state population size and distribution mimic reality, users can more confidently attribute this to model design and calibration.

Users must remember to differentiate between transient dynamics and historical trends. If an actual population is declining due to ongoing habitat loss, then there may be data available illustrating the change in population numbers through time. If a HexSim simulation mimicking this population is started with an artificially large population, it too will exhibit a sharp decline. It may be tempting to compare the two rates of decline, and argue that similarities suggest a good model fit. To do so would be wrong. The actual population was declining due to ongoing habitat loss, whereas the simulated population was declining due to initial conditions. In this example, one cannot even say with confidence that the steady-state has been achieved when the simulation population size meets today's actual population size. Transient dynamics can include under-shoots and over-shoots, and sometimes take a long time to die away completely.

If users want to do better, they can perform hindcasting, but this can be a great deal of work, and the spatial data necessary to so is often not available. If forecasting is the goal, then users will often have developed one or more spatial data time series that illustrate future conditions. These future landscape changes can be initiated starting at a point in the simulation where it is clear that the start-up bias has disappeared. Then model calibration will entail making sure that the simulated steady-state conditions match current population size, spatial distribution, and trait distributions prior to the initiation of landscape change.

Initial Trait Distribution

Every HexSim trait has rules that dictate what values will be assigned to members of the initial population. These rules vary by trait type. The distributions of trait combinations across a population is an important measure of the population state, and potentially quite useful for model calibration. But there is little value in trying to initialize a simulation with realistic distributions of trait values. This would be hard to do correctly, and it is more important to provide initial trait combinations that keep the population

extant until steady-state can be achieved. For example, suppose all non-zero reproductive rates are associated with late-stage individuals in a stage-stratified population. Initializing such a population with only stage-zero individuals might prevent it from ever achieving a positive growth rate. This is critical because it means an otherwise correctly parameterized simulation may appear completely unrealistic due strictly to a poor choice of initial trait values. Just as was the case with the initial population size and distribution, the initial trait distribution should facilitate the model calibration process. These parameters should be given strategic initial values, as opposed to realistic ones.

Hindcasting

Hindcasting is an approach to calibration that involves starting a model with past conditions so that a simulation advances to the present. When contemporary events or dynamics can be recreated, this provides strong evidence that the simulation design and parameter values are accurate. In HexSim, hindcasting will typically involve the development of spatial data time series that capture landscape change from the past to the present. Also, spatial and non-spatial disturbance regimes might have to be developed to capture fire regimes, disease outbreaks, hunting pressure, urban growth, etc. A simplifying assumption that can usually be made safely is that the species' life history is static, and only the landscape conditions and disturbance regimes need change. Users would run such a simulation to steady-state with the initial landscape and disturbance data. Once steady-state was reached, the progression of landscape and disturbance regime change would be initiated. When the simulation reached the present day, the population size, distribution, and structure should agree with contemporary data if the model was calibrated correctly.

Because today's population size and distribution are often a consequence historic disturbance regimes and past changes to landscape structure, it may be impossible to recreate current population trends without hindcasting. That is, an actual population may not have achieved steady-state. It may instead be exhibiting transient dynamics due to the time series of stresses it has encountered in the real world. These dynamics cannot be recreated by running a simulation to steady-state with a static landscape and fixed disturbance regime. Such a simulation can really only hope to shed light on the landscape carrying capacity over the long term. For this reason, a combination of hindcasting and forecasting will be ideal in situations where policy alternatives are to be evaluated. Hindcasting would allow contemporary population dynamics to be recreated, and the forecasting component could quantify population trends likely to result from management actions. The additional realism that resulted from hindcasting would add credibility to the forecasts, and would make it possible to assign a meaningful time-line to the simulation results.

Landscape Dynamics

Outside of parks and preserves, there is little that is static about landscape conditions in the modern world. HexSim has been designed to make it easy for users to add landscape dynamics to their simulations, with as much or as little detail as the problem at hand dictates. But data on landscape change can be hard to come by, and in many cases landscape change may have to be ignored for practical reasons. This need not limit HexSim's utility as a tool for evaluating management alternatives, recovery strategies, and so on. The most reliable products from a simulation model like HexSim are predictions of *relative* population size and trends. When generating measures of relative change, inaccuracies will often cancel. The least reliable products from a complex population model are going to be the exact measures of population size or distribution. Our inability to precisely characterize environmental stochasticity, as just one example, illustrates this limitation. Changing the sequence of good and bad years can have a huge impact on the population size at any given time. But the details of environmental stochasticity may have no impact on the relative ranking of four competing recovery strategies attempting to balance conservation efficacy with cost. Thus what is required to justify arguing that a model has been adequately calibrated is going to depend on its application. When data on landscape change are unavailable or limited, this places limitations on what can be expected to be gained from the simulation results. But these limitations may not be severe. And because the model is relatively simpler, it should be easier to understand and calibrate.

Changing Disturbance Regimes

Disturbance regimes can be significant drivers of population dynamics, and many are in play simultaneously in the real world. HexSim has the ability to simulate multiple interacting (or independent) disturbance regimes, and in fact this may be the model's most distinguishing feature overall. Disturbance regimes in HexSim typically modify the traits of affected individuals, and this in turn alters their behavior or vital rates. But users must quantify the impacts of any simulated disturbance. Adding disturbance regimes makes a model more complex and harder to calibrate. Never the less, doing so may be critical, and there are ways to balance complexity with realism. For example, not all disturbance need be simulated mechanistically. Hunting pressure might be added using a probabilistic trait. If victims die with certainty, then only a single variable -- the probability of being shot -- might need to be added to a simulation. Additional realism could be gained by varying the hunting pressure spatially. If three levels of hunting pressure were simulated, then three rates would have to be supplied instead of one.

A more complex example could link resource acquisition and disease. Individuals might move around a landscape in search of food. To achieve a high resource acquisition trait value they could be required to both be in a resource-rich location, and out-compete conspecifics who have also arrived there. The disease might be transferred from individual to individual as they interact while exploring for resources. Individual survival rates could vary depending on both disease status and resource acquisition class. The rates of recovery from the disease might also be affected by resource acquisition history. And an individual's competitive ability could be lowered by the disease. All of this is straightforward to build into a scenario, but doing so adds complexity and makes

calibration more difficult. As described above, users need to strike a balance between model realism and complexity. More realistic simulations can be used to generate more precise predictions, but only if they are adequately calibrated. And calibrating a complex model is a bigger challenge. Simpler simulations will be easier to calibrate, but users will have to grapple with the implications of limited model realism. The conclusions drawn from simpler simulations may be less certain, or may be limited to statements about relative change, etc.

Simplifications and Omissions

HexSim simulations will always be dramatic simplifications of complex dynamic processes. And the same is true for all models of populations and ecosystems. An important principal in developing HexSim simulations is that enough complexity should be added to capture the important drivers of population dynamics, but no more. Users should be aware of what simplifications are being made, and which aspects of real world complexity have been omitted entirely. Recognizing these simplifications and omissions will help to determine what conclusions can be safely drawn from the model results. This will in turn help elucidate model calibration goals. For example, if environmental stochasticity is left out of a model, then it may not be credible to use it for generating probabilities of extinction. But it may still be reasonable to use the scenario to form estimates of the mean steady-state rate of population growth. Estimating of the probability of extinction might also entail adding additional realism, such as separately modeling both sexes so that an Allee effect can be simulated. And this will necessitate adding mate finding and other complexities. Thus divergent strategies can emerge for model development, and for model calibration. The selection of which competing model to use would depend on the research products the simulation was expected to generate.

Sensitivity Analysis

Part of the calibration process should involve a comparison of parameter sensitivity and parameter uncertainty. No parameter will every be known exactly, and sensitivity analysis can provide users with an idea of how much the uncertainty in a parameter matters to the simulation results. If a simulation is highly sensitive to a poorly known parameter, then that uncertainty should be addressed explicitly by gathering results from a range of parameter values. On the other hand, uncertainty in some parameters will have little significance for the simulation results. HexSim has a Sensitivity Analysis utility that can help users perform these analyses.

Model Performance

The principal products of a HexSim simulation are population trends, population distributions, and trait distributions for each population. These simulation products integrate all of the parameter values and interactions built into a scenario. If the model design captures the critical life history traits, stressors, and other features, and if it has been carefully calibrated using available data sets, then the simulation results should be meaningful. Still, mistakes can easily be made, and plausibility testing should be employed to help identify errors. This section briefly discusses plausibility testing in the context of the simulation products mentioned above.

Population Trends

In most cases, HexSim populations will exhibit one of the six general types of behavior listed below:

- Asymptotic growth
- Asymptotic decline
- Population collapse
- Quasi-stable equilibrium
- Cyclic behavior
- Chaotic behavior

Asymptotic growth will be exhibited when populations have an initially positive growth rate, but settle into a steady state as they reach the landscape's carrying capacity. It is possible to develop scenarios without density dependence that exhibit unbounded growth, but such a simulation would be unrealistic.

Asymptotic decline refers to population decline to a nonzero steady state. Often populations will persist at low levels in the best habitat fragments. They will tend to be highly sensitive to stochastic extinction, but otherwise can hang on for extended periods of time in select portions of the landscape.

Population collapse simply refers to an unbounded population decline leading to extinction.

In the absence of disturbance, a population with a positive growth rate will not settle into a steady-state below carrying capacity. But constant, or even periodic disturbance can produce such a result. Likewise, in the absence of immigration, a population with a consistently negative growth rate will eventually decline to extinction. But regular introductions of new individuals can create a stable nonzero population size. These

conditions are referred to as a quasi-stable equilibrium since they only exist as a result of external forcing.

Populations that exhibit cyclic behavior achieve stability without ever reaching a steady-state. Cyclic behavior will often result from population interactions, such as predator-prey dynamics.

Chaotic behavior is meant to characterize population trajectories so variable that no clear trend emerges. In some cases, a trend may become apparent if the simulation is run for a sufficient number of time steps.

Asymptotic growth, asymptotic decline, and population collapse are closely related. The other three types of trends are quite different. But more than one behavior may be exhibited by a population simultaneously.

With HexSim, it is easy to generate long time series of simulated population data, and trends quickly become apparent. In reality, we often have just a small number of data points to work with, and it can be difficult or impossible to extract trends from the available information. Users should ask themselves if the trends exhibited by HexSim are plausible, given the real-world data that are available. Simulations with a range of complexity should ideally be compared, to see if they all exhibit the same characteristic population trends. If they do not, then one must explore why and how the additional complexity fundamentally altered the population dynamics. It may not be possible to know which model variant is the most accurate. Likewise, sensitivity analysis should be conducted to examine whether population trends undergo characteristic shifts when parameters change by small amounts. These types of investigation will help users calibrate their confidence in the simulation results. They will also point the way towards fruitful areas for future research.

Population Distributions

The simulated distribution of individuals across a landscape can provide a powerful tool for testing a scenario's plausibility. Population distributions are influenced by population size, by rules for habitat use and space requirements, by movement behavior, disturbance dynamics, and many other factors. And it's hard to correctly reproduce population densities across large complex landscapes using a flawed model. HexSim also makes it easy to simulate population surveys, so data on simulated densities can be gathered in a pattern that matches the actual information collected in the field. Difficulties arise because existing field data are often insufficient to discern how population densities really are changing from place to place. Further, population densities fluctuate in time, and survey methodologies are not always consistent. In spite of these complications, simulated population distributions should be used to help evaluate model plausibility, to the extent possible, since they provide such a powerful integrative measure of the simulation's mechanics.

Trait Distributions

The use of trait distributions for plausibility testing will always be difficult. For one, many HexSim traits will be categorical representations of continuous real-world variables (e.g. fitness). Further, little will likely be known about the actual population attributes that might be characterized as traits. On the other hand, HexSim is flexible enough that traits can be tailored to match population attributes that have been measured. Trait value frequencies are gathered using Census events and tallies, and can be displayed dynamically with the Simulation Viewer. This data can be extremely useful for debugging a scenario. Census data will point out when individuals are acquiring too few or too many resources, or when exposure to a stressor is unrealistically high or low, etc. In other cases, trait distributions might illustrate population dynamics that are not evident from measures of population size or distribution. For example, the distribution of genotypes, or disease status may vary spatially, and these patterns might be of interest in a study. A comparison of the simulated and expected or observed trait distributions can reveal a great deal about the model behavior that cannot be learned any other way.